AFRL-SR-AR-TR-04-

0370

REPORT DOCUMENTATION PAGE

1. REPORT DATE *(DD-MM-YYYY)*

2. REPORT TYPE
Final

3. DATES COVERED *(From - To)*
1 September 2003 - 31 May 2004

4. TITLE AND SUBTITLE
Detecting and Mining Similiarities, Differences and Target Patterns in Sequences of Images using the PFF, LGG and SPNG Approaches

5a. CONTRACT NUMBER

5b. GRANT NUMBER
F49620-03-C-0091

5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

6. AUTHOR(S)
Mrs. Despina Bourbakis

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Automation, Integration of Information & Systems Inc.
9834 Counrty Creek Way
Centerville, OH 45458-9244

8. PERFORMING ORGANIZATION REPORT NUMBER

20040721 047

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research
4015 Wilson Blvd
Mail Room 713
Arlington, VA 22203

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution Statement A. Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
In phase I the identification and significance of the problem was the mining images, especially sequences of images or video for detecting-extracting, fusing and recognizing differences, changes and associating patterns. These types of problems are difficult challenges in the image analysis and computer vision research community. These difficulties mainly due to the textural nature of the images and the possible noisy conditions during their capture. The recognition component was not a part of the phase I, but it helongs to phase II. We did it, however, in phase I in order to focus in phase II on the integration and real-time issues. Thus, for the achievement of the phase I tasks (objectivcs), wc have developed and /or used several methods such as Pixel Flow Functions (PFF) (or projections), Segmentation, Local Global Graphs (L-G), Genetic Algorithms (GAs). Registration (or Mapping), Curve Fitting. Wavelets, Region Synthesis, Stochastic Pctri-Nets (SPNs), and others. The efficient uses of these methods in a certain sequence has produced the desirable results for each of the tasks. Here we present each task and the sequence of methods involved for obtaining the results

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Despina Bourbakis
					19b. TELEPHONE NUMBER *(Include area code)*

BEST AVAILABLE COPY

From : Automation, Integration of Information & Systems Inc.
Mrs Despina Bourbakis, President

To: Dr. Robert Herklotz and Mr. Stanley Borek

Detecting and Mining Similarities, Differences and Target Patterns in Sequences of Images using the PFF, LGG and SPNG Approaches

Final Technical Report

In phase I the identification and significance of the problem was the mining images, especially sequences of images or video for **detecting-extracting, fusing and recognizing differences, changes and associating patterns.** These types of problems are difficult challenges in the image analysis and computer vision research community. These difficulties mainly due to the textural nature of the images and the possible noisy conditions during their capture. The recognition component was not a part of the phase I, but it belongs to phase II. We did it, however, in phase I in order to focus in phase II on the integration and real-time issues. Thus, for the achievement of the phase I tasks (objectives), we have developed and /or used several methods such as Pixel Flow Functions (PFF) (or projections), Segmentation, Local Global Graphs (L-G), Genetic Algorithms (GAs), Registration (or Mapping), Curve Fitting, Wavelets, Region Synthesis, Stochastic Petri-Nets (SPNs), and others. The efficient uses of these methods in a certain sequence has produced the desirable results for each of the tasks. Here we present each task and the sequence of methods involved for obtaining the results:

Task-1: Detecting and Extracting Differences and Changes in Sequence of Images
(PFF, L-G graphs, Registration, GAs, Segmentation)

Task-2: Fusing Visual and Thermal Images
(Registration, GAs)

Task-3: Tracking Targets and SPN Formations of Patterns
(PFF, L-G graphs, Segmentation, Registration, SPNs)

Task-4: Recognizing Patterns or Objects
(Curve Fitting, Wavelets, Segmentation, Region Synthesis, L-G graphs)

TASK-1: DETECTING AND EXTRACTING DIFFERENCES AND CHANGES IN SEQUENCES OF IMAGES

1. Introduction

In this work we handle the problems of image registration, change detection, object/target tracking and multimodal fusion in the domain of aerial imagery. This domain is attracting a constantly growing interest and it has many applications, including monitoring of urban development and environmental changes, target detection and tracking, medical imaging applications and is also closely related to motion estimation and general video processing methodologies.

Image and video registration is the process of geometrically aligning an image pair of the same scene taken under different viewpoint, illumination and temporal conditions. Several methodologies have been proposed including correlation-based, Fourier domain and feature

matching approaches. Following the alignment of the two images, a change detection method is applied that detects the meaningful differences between the image pair. These methods are also used for motion analysis and especially in motion segmentation to extract the moving objects in dynamic scene analysis i.e. object tracking. Usually change detection is employed to reduce the amount of data for further processing. Furthermore in the context of aerial imagery the visual information derived from several optical sensors operating in varying wavelengths and having different resolutions. An important problem in this context is the fusion of information from different sources. In this work we consider these problems in a common framework to build an aerial vision scheme.

Registration Methodologies- As mentioned above, the literature in this subject is very rich and several methods of dynamic scene analysis can be used for this purpose [3]. Earlier attempts were based on template matching using statistical measures for example cross-correlation, cross-covariance, absolute differences etc. These methods are adequate for simplified conditions, i.e. for translational variations of the image scene. On the other hand, correlation-based measures are not so efficient for images corrupted by other types of noise or illumination differences. Another group of methods employs spatial projections on horizontal, vertical or arbitrary oriented axis for registration and motion estimation [4,13]. These methods are also formulated by the Radon transform and can address translation and rotation transforms. Several Fourier domain methods were also proposed [9,12], based on the Fourier Amplitude or Phase correlation and the Fourier Mellin transform mainly. According to these approaches, some properties of the Fourier transform related to the Shift theorem are used to account for translation, rotation and scale differences. In addition to that, they perform well in the occurrence of frequency dependent noise and handle the illumination changes and variations due to different sensors efficiently. Another advantage of these methods is that they can be implemented fast using *FFT*'s. However, a usual problem of Fourier-based techniques is the aliasing effect that can be resolved using windowing operations. Furthermore, these methods are limited to specific well-defined transformations, mainly translation and rotation. When a spatially local transformation exists (temporal registration) or when the images contain different parts of the scene, these methods present shortcomings .Local variations can be efficiently handled by optical flow-based techniques [8,11]. The optical flow can be defined as the solution of the differential equation of the motion constraint and the main solutions to it were proposed by Horn and Schunk, and Lucas Kanade [11]. These methods are regularly applied to motion estimation and segmentation applications. They can address translation, rotation and scale, for relatively small displacements and they suffer from the aperture effect. Hierarchical versions of these methods can be applied for larger displacements [2]. Another popular category is represented by the feature-based methods. These mainly use point mappings to address more complicated variations. In general, the point mapping approach can be formulated as follows. Given a set of points R in the reference and another set I in the misaligned (input) image, our task is to determine the optimal spatial transformation in the space T that maximizes the similarity metric $D_{R,I}$, which is defined in the space of Cartesian product of R and I, $D_{R,I} \in RxI$.

Change detection [10,15]is closely related to registration i.e. a good registration result simplifies this process. These approaches may be generally classified into pixel-based and region-based. The simplest approaches are based on pixel differences and a subsequent thresholding operation to detect the significant changes. Other more recent approaches select the threshold by estimating the noise of the difference image, or by using statistics of the background pixels. Nevertheless these methods are sensitive to noise and radiometric changes. Therefore some region-based approaches were proposed, which estimate characteristics over image areas. These methods include hypothesis testing, or likelihood ratio testing approaches. However these methods are still illumination dependent, so shading and texture models were proposed to achieve illumination invariance.

The proposed method- Here we developed a framework that deals with the correspondence problem for aerial images. The requirements are to have an automated registration approach and efficient change detection. Moreover, fusion of different sensors should be feasible and the method should be temporal and illumination invariant. The registration is completed by a stochastic optimization scheme using Genetic Algorithms to determine the Affine Transform coefficients to match the image pair. Next to that, we employ a hierarchical optical flow approach to account for the smaller displacements of the image and improve the change detection effectiveness. This simplifies the change detection process that is completed using the differencing and thresholding operation.

2. GAs Optimization

In this paragraph are explained the main steps of GA-based registration. A point mapping scheme is presented that uses stochastic optimization to automate the feature mapping process. At this stage the employed feature space is defined by the pixel intensity. The proposed scheme is outlined in figure 1. The search space corresponds to the conformal mapping or the affine transformation, which are adequate for aerial images. In most cases the conformal mapping (also known as rigid transformation) approach is sufficient.

Affine transform. In the case of affine transform a point (x_1,y_1) of the input image is mapped onto the point (x_2,y_2) of the registered image as expressed by the following equation:

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} a & b \\ d & e \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} c \\ f \end{bmatrix} \qquad (1)$$

The parameters *a-f* are the affine parameters and they cover the several forms of misalignments i.e. translation, scale, rotation and shear. Affine transforms are linear since they preserve straight lines.

Conformal mapping. When the shear operation is omitted the corresponding transform is called conformal mapping and equation (1) becomes:

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} r \cdot \cos\theta & r \cdot \sin\theta \\ -r \cdot \sin\theta & r \cdot \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \qquad (2)$$

Here, r expresses the scale, θ the rotation, and t_x and t_y the translation parameters. This case is illustrated in figure 2.

Generally, in the point mapping process the most important and difficult step is feature selection. This is due to the occurrence of outliers, digitization effects, noise and illumination variations that affect the feature values. Although several feature selection and extraction methods have been proposed, this task becomes difficult when significant amount of uncorrected variations occur and it remains an open research topic.

*Automation of the process using feedback-*In order to overcome the above difficulty, a feedback scheme may be employed to define the optimal transformation. Some previous related methods include relaxation, clustering and hierarchical search approaches; however they were limited to simple variations or implied impractical computational complexity. In our scheme the genetic algorithm optimization approach [7] is used to find the optimal registration parameters in the global search space. The employed search space corresponds to the class of the conformal (or

rigid) spatial transforms. The feature selection process is not necessary in this approach; the overlapping areas of the examined images are compared instead. This process is described next.

The Genetic Algorithms approach is a stochastic method, very well suited for optimization problems in several fields [7,8]. The initial search space is reduced using principles of the evolution theory, in order to maximize a user-defined fitness function. The input variables of the genetic algorithm, known as chromosomes are bit-strings that contain the parameters of the spatial transformation. The population of chromosomes is selected randomly in the initial search space and a fraction of the good solutions is selected while the rest is eliminated. The chromosomes that remain are combined according to the three basic operations of reproduction, crossover and mutation. More specifically, reproduction is the operation by which the strings that produce high fitness values remain in the next generation. Crossover is the random combination of the best strings coming from the previous generation. Reproduction and crossover are responsible for the searching capability of genetic algorithms. The mutation operation prevents the genetic algorithms from converging to a local solution by randomly changing the binary value at a location in the bit string. The above process is iterated until the algorithm converges to the final solution within a generation.

According to the previous paragraph, a fitness function is evaluated for each chromosome of the population, and should be maximized. Since our goal is to achieve efficient image registration, a robust similarity measure has to be defined in order to express the similarity of the transformed image with our reference image. Several measures were tested, including the correlation, covariance, correlation coefficient and sum of absolute differences. It was concluded that the centered sum of absolute difference produced more efficient results than the other measures. This measure is expressed by the relation:

$$D_{R,I} = \sum_x \sum_y \left| R(x,y) - \hat{R} - I(x,y) + \hat{I} \right|, \tag{3}$$

where R and I are the reference and input images while $\hat{R}$ and $\hat{I}$ are the mean estimates respectively, calculated over the overlapping area of the image pair. This function is inverted to function as a fitness measure. In the following paragraph are reported some comparative experimental results of this scheme.

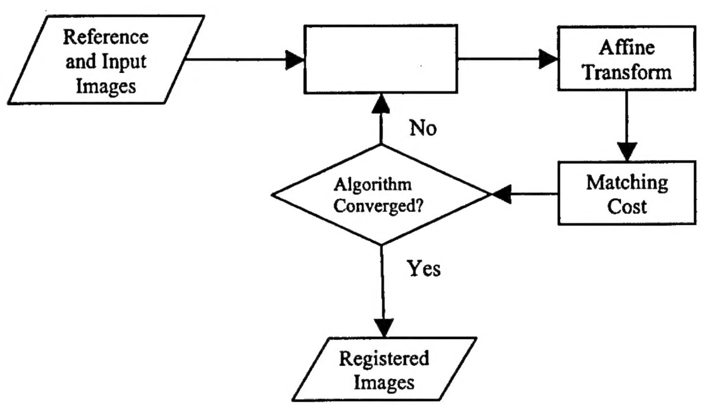


Figure 1. The feedback optimization registration scheme.

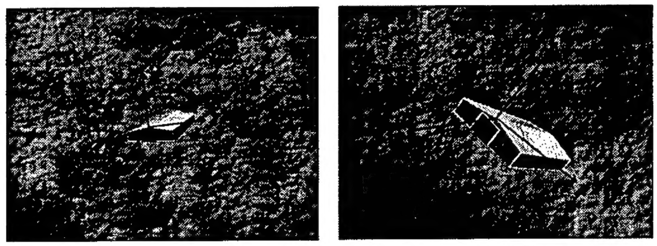


Figure 2. A synthetic example of two aerial images that contain scale, rotation and translation variations.

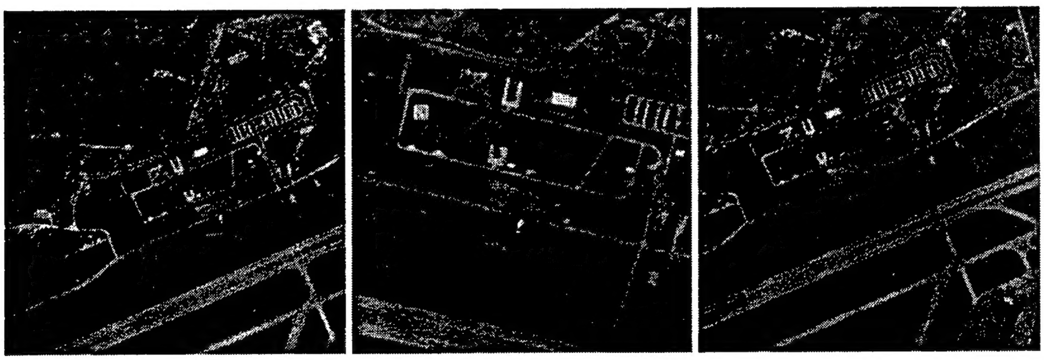


Figure 3. The original test images.

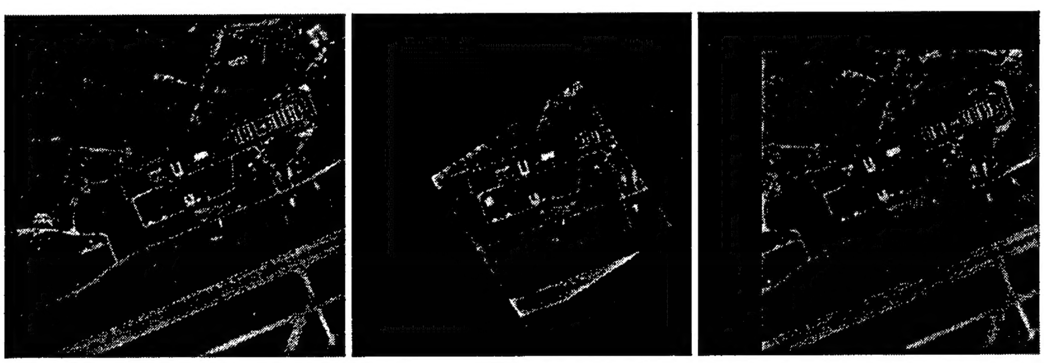


Figure 4. Registered images.

3. Hierarchical Optical Flow

In the computer vision field it is essential to compute the image motion. This is defined as the perspective projection of the 3D scene points that move relative to a camera on to the 2D imaging surface [8]. The optical flow approximates the image motion under the condition that the illumination changes are caused by motion, and the surfaces of the scene are Lambertian. Apart from that the image motion can be estimated by the correspondence of some features points that are usually corner points or spatio-temporal homogeneous image regions and estimate the motion vectors. It was indicated that for the case of slow motion the methods of the first category, also called differential methods, produce more accurate results than correspondence methods.

In this work we use the differential method similarly to Lucas and Kanade [11] in a hierarchical scheme for the case of affine motion. In general, the motion constraint equation is expressed as follows:

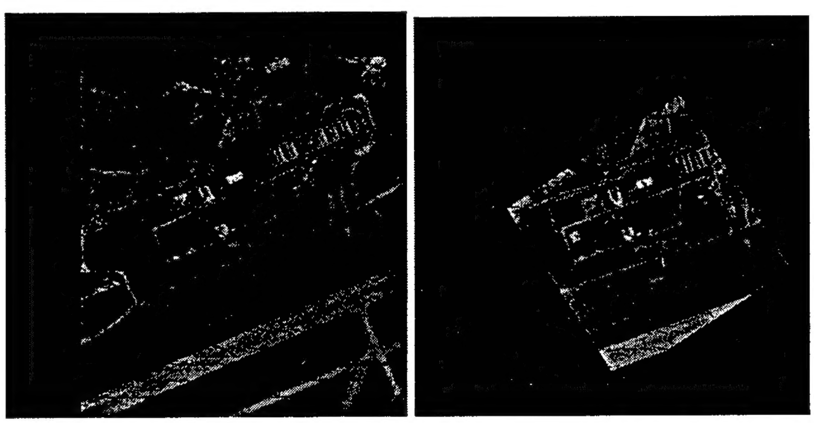
$$\nabla I \cdot \vec{v} = -\frac{\partial I}{\partial t} \Leftrightarrow I_x \cdot v_x + I_x \cdot v_y + I_t = 0\,, \tag{4}$$

where I is the image intensity, I_x, I_y and I_t are the spatial and temporal derivatives of I and $\vec{v} = (v_x, v_y) = (\frac{\partial x}{\partial t}, \frac{\partial y}{\delta t})$ is the velocity vector. This equation holds following the assumptions of locally translational motion, preservation of intensity over time and the continuity of the image over space and time. A well known ambiguity pitfall known as the aperture problem is overcome by assuming the constant velocity constraint and the image velocity is calculated in the least squares sense after calculating the spatial and temporal derivatives for a set of points.

For the case of large displacement, a hierarchical framework is followed that includes the following stages:

1. A Gaussian image pyramid is constructed,
2. Starting from the coarsest towards the finest resolution,
    3. The affine motion is estimated,
    4. The image is iteratively warped,
5. Method is completed after the finest resolution is processed.

This approach is efficient when applied to the registered image the refine the alignment prior to the change detection process.



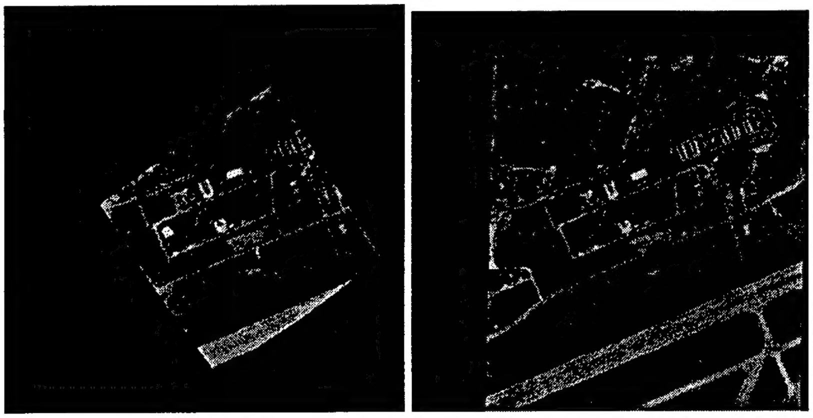


Fig. 5. Optical flow fields (first row) and the aligned images using hierarchical optical flow (second row).

4. Change Detection

The fine alignment process facilitates the change detection process. Two approaches were adopted. According to the first one simple differencing is estimated first and a thresholding operation follows to produce the final differences. In figure 6 are displayed the image differences before and after the fine alignment. From these images it is obvious that the hierarchical optical flow estimation is critical for the final results.

A more robust process is to estimate the displaced frame differences and apply segmentation to the first image to estimate the regions activity. The regions with the most active pixels are classified as different. The authors are currently working on the implementation of a texture-based, illumination invariant measure to be included in the change detection process.

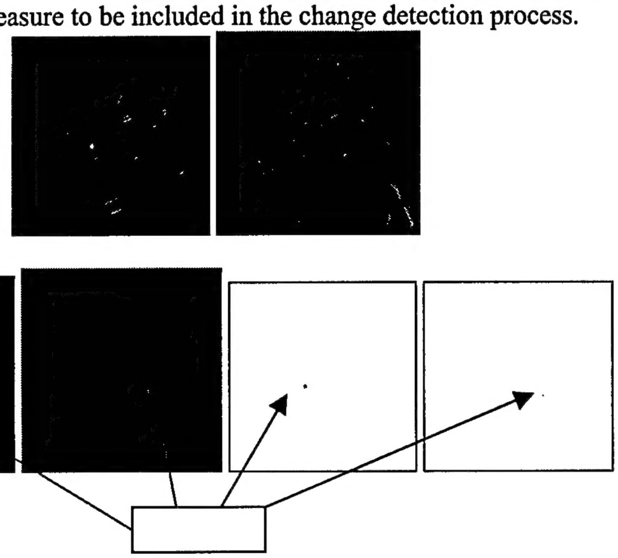


Figure 6. Differencing before alignment (first row) after alignment (second row) .

5-Alternate Approaches

5.1- Merging PFF+LGG for detecting changes in sequences of images

At the AIIS site the researchers are working for the completion of the merge of the Pixel Flow Functions (PFF) method with the Local Global Graph (LGG) method for detecting changes and patterns in sequences of images.

The PFF method is a sensitive method on changes that occur at the pixel level. When a video is taken, however, the PFF method as it is (sensitive) cannot perform well because detects all the changes that occur between two MPEG lossy compressed images. More specifically, two non consecutive MPEG frames carry differences at all of their pixels due to different illumination and lossy compression on these pixel values. These differences are not necessarily visible to human eye. In order to merge the PFF method with the LGG method, an appropriate threshold has to be defined for the PFF sensitivity. This threshold with allow us to automatically frame the differences as shown in the figures 7, below. In particular, many image frames were captured by a digital camera (video). From the video images 4 non consecutive frames were selected for testing the PFF and LGG algorithms. The reasoning for selecting non consecutive image-frames is that, this way tests the robustness of the examined methods and is a challenge as well. More specifically examining consecutive frames, it is easy to find something that was not well defined in a previous frame, and the tracking process is simpler. Having, however, non consecutive image frames to examine, the challenge is greater than the consecutive frames case, since we may know very little about previous frames. This represent the case of a very difficult problem, where there is a moving camera and moving targets as well.

In the presented test here, the image frames contain people walking in two different groups, see frames around these groups, figures 7 and 8. The frames were extracted from these images and a Region Growing based on fuzzy like reasoning was performed on these frames for determining the regions of change (or difference), see the figures 8, below. The LGG method was performed on the segmented images for determining the regions (L-G) graphs.

The LGG method is used to extract the regions and the shape of the detected differences. For this particular part of the task-1, we used two different image segmentation techniques, the Region-Growing and the Watershed with Clustering. The reasoning behind this effort is to ensure that the extracted regions are accurate with minimum loss of information for the later synthesis and recognition of the patterns and targets.

5.2- The Image Segmentation Approach and Changes Extraction

In order to extract the objects inside the Rectangles of Displacement some other edge and region based segmentation methods were considered as well using watershed segmentation, graphs and clustering and fuzzy logic approaches. In the following figures 9 some results are displayed for watershed based segmentation, which provides improved delineation accuracy. Apart from that a small study was conducted comparing our results in different color spaces.

The Region Growing and the Watershed with Clustering segmentation methods show different results. Since our goal is to synthesize different image regions, we have to study which of these two techniques (or a combination of them) is the best way for selecting the correct image regions for synthesis and later recognition of the "targets or patterns" composed by these regions. This particular effort will continue and a good feedback is expected during the fusion task of this project.

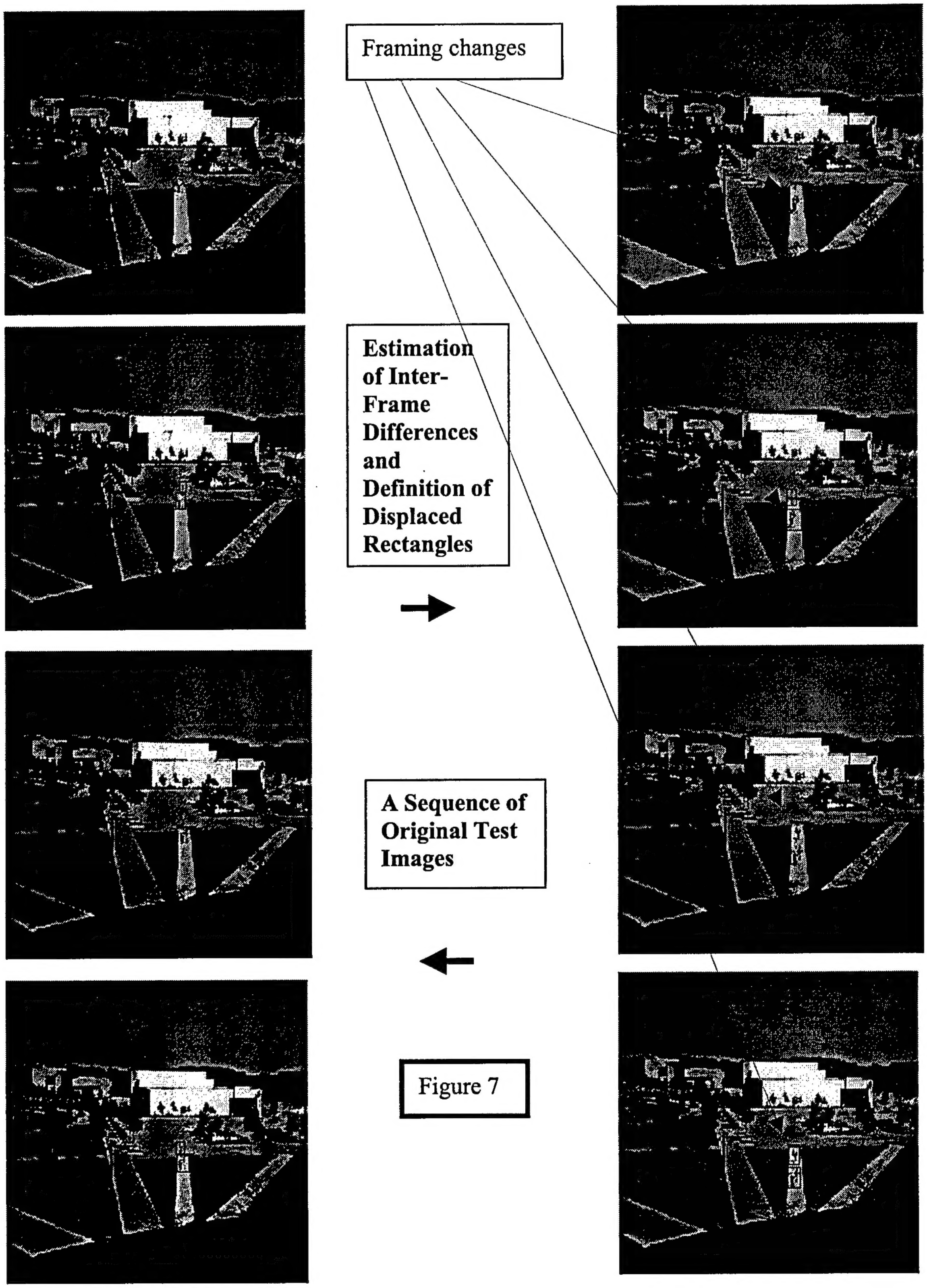

Framing changes
Estimation of Inter-Frame Differences and Definition of Displaced Rectangles
A Sequence of Original Test Images

Figure 7

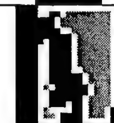
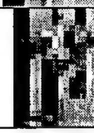
Frame no. 1	RoD[1]	L-G graph	RoD	L-G graph
1				
2				
3				
4				

Figure 8: Segmentation and Application of L-G approach on the Displaced Regions

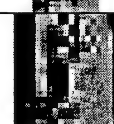
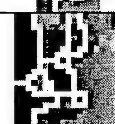
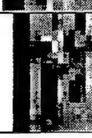
Frame no. 1	RoD	Alternate approach	RoD	Alternate approach
1				
2				
3				
4				

Figure 9: The alternative segmentation and the generation of the region graphs

References

[01S. Alliney and C. Morandi, "Digital image registration using projections", IEEE Transactions on Pattern Analysis and Machine Intelligence., vol. 8, no. 2, pp. 222-233, 1986.

[02]L.Barron and M.Khurana, Determining optical flow for large motions Using Parametric Models in a Hierarchical Framework, Proc. of Vision Interface, Kelowna, B.C., pp. 47-56, 1997.

[03]L. G. Brown, "A survey of image registration techniques", ACM Computing Surveys, vol. 24, no. 4, pp. 325-376, December 1992.

[04]S.C. Cain, M.M. Hayat, and E.E. Armstrong, "Projection-based image registration in the presence of fixed-pattern noise", IEEE Transactions on Image Processing, vol. 10, no. 12, pp. 1860-1872, 2001.

* The video was taken by Dr. N. Bourbakis from the Engineering building at Wright State University.

[05] P. Chalermwat and T. El-Ghazawi, "Multi-resolution Image Registration Using Genetics", Proceedings of IEEE International Conference on Image Processing (ICIP 1999), 24-28 October 1999, Kobe, Japan, vol. 2, pp. 452-456.
[06]Y. Chen et al, "Efficient Global Optimization for Image Registration", IEEE Transactions on Knowledge and Data Engineering, vol. 14, no. 1, January/February 2002.
[07]D. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley, New York, 1989.
[08] B. K. P. Horn, Robot Vision, the MIT Press, 1986.
[08] C.D. Kuglin, D.C. Hines, "The phase correlation image alignment method", IEEE Conference on Cybernetics and Society, pp. 163-165, 1975.
[09] L. Li and M.K.H. Leung, "Integrating Intensity and Texture Differences for Robust Change Detection", IEEE Transactions on Image Processing, Vol. 11, No. 2, February 2002.
[10] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision", Proceedings of the 7th International Joint Conference on Artificial Intelligence, Vancouver, pp. 674-679, 1981.
[11] B. S. Reddy and B. N. Chatterji, "An FFT-based Technique for Translation, Rotation and Scale-Invariant Image Registration", IEEE Transactions on Image Processing, vol. 5, no. 8, pp 1266-1271, August 1996.
[12] A. A. Cole-Rhodes, K. L. Johnson, J. LeMoigne, I. Zavorin, "Multiresolution Registration of remote Sensing Imagery by Optimization of Mutual Information Using a Stochastic Gradient", IEEE Transactions on Image Processing, vol. 12, no. 12, pp. 1495-1511, December 2003.
[13] D. Robinson and P. Milanfar, "Fast local and global projection - based methods for affine motion estimation", J. of Mathematical Imaging and Vision, vol. 18, pp. 35-54, January 2003.
[14] N. Rowe, and L. Grewe, "Change Detection for Linear Features in Aerial Photographs Using Edge-Finding", IEEE Transactions on Geoscience and Remote Sensing, vol. 39, no. 7, pp. 1608-1612, July/August 2001.
[15] Q. Zheng and R. Chellappa, "A computational vision approach to image registration", IEEE Transactions on Image Processing, vol. 2, no. 3, pp. 311–326, July 1993.

TASK-2 : FUSING VISUAL AND THERMAL IMAGES

In several cases our objective is to fuse and align images acquired from different sensors, such as visual and thermal. In these cases the relationship between the intensities of the examined image pair is unknown and it usually depends on the environmental conditions. As pointed out in previous works (for example in [1.2]) two main considerations have regularly to be addressed; i) the image representation and ii) the similarity measure.

In this part of our project research an attempt was made to develop an illumination invariant approach that will facilitate the image matching process. Some usual representations are based on contour features, vector fields and feature points. Although their being promising, most of these methods include thresholding operations and feature tracking processes which have not converged to robust results under varying conditions.

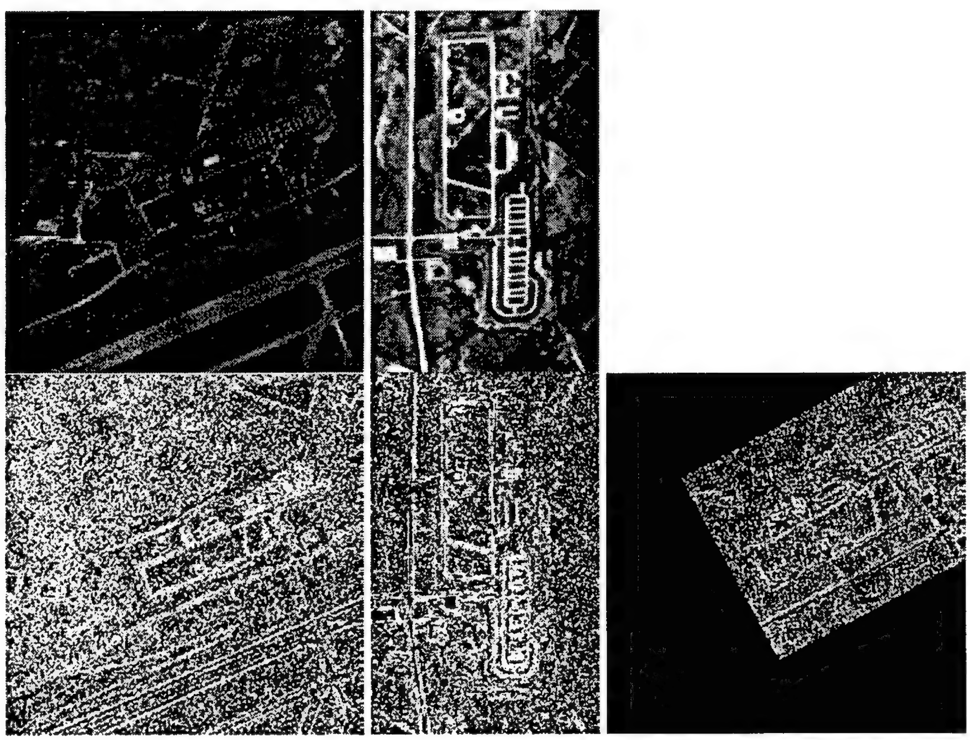
The method employed here, was inspired by the observation that the main image feature that remains relatively unchanged under multimodal imaging is the texture. The texture is mostly evidently in finer resolutions, where high frequency information is present. In coarser resolutions usually the texture is smoothed out and the edges between different objects become more apparent [2]. It has also been indicated that the human visual system implies multiscale processing in its operation. This idea has also been used in the fields of scale space and wavelet processing for image segmentation [3] and object recognition schemes.

In this report the authors have tested three different representation approaches that were included in the GA-based image registration scheme; the first is the operator Laplacian of

Gaussian that contains high frequency information, the second was derived from a non linear diffusion process and the third is based on non-parametric edge detection scheme.

The Laplacian of Gaussian (LoG) is a well known operator that has been used for edge detection and localization. This operator is the result of the application of the second derivative operator on the Gaussian kernel function and the computation of the energy of this signal. This method produces acceptable results; however it is not efficient for cases that include rotation misregistration. This mostly attributed to the fact that the LoG is rotation invariant. In the next stage the authors developed a texture representation scheme based on the idea of non linear diffusion. According to this approach the image is smoothed in the areas of homogeneous intensity while preserving the location of significant edges as it was first proposed by Perona and Malik [5]. This operation can be also regarded as non-linear smoothing that uses an adaptive Laplacian operator to detect the homogeneous intensity areas. The texture areas now are extracted by comparing the original image, with the diffused counterpart. The absolute differences are readily estimated, to extract the texture information. The third representation implies a non-parametric edge-detection method using Parzen kernels [4]. This estimates the edge location and it was concluded that it produces better registration results compared to the other two representation schemes.

Based on this representation, the image registration is carried out by means of GA optimization and the final results are refined using multiscale optical flow as described in previous sections.



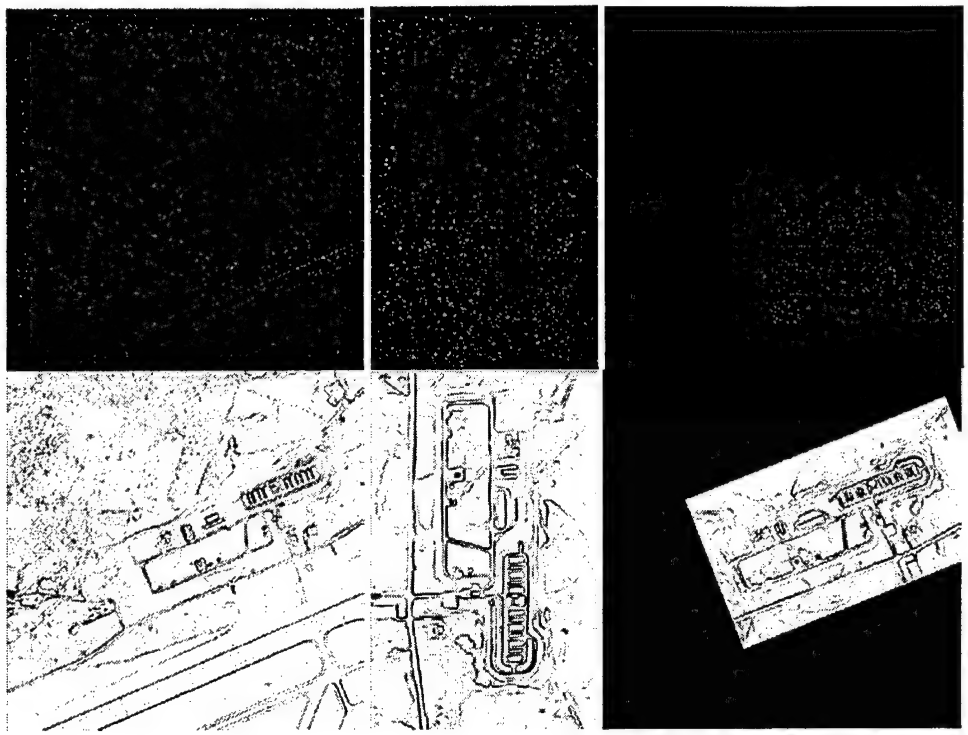


Figure 1: An example of a visual (first row-left column) and a thermal image (first row-right column) of a similar scene. The LoG energy representation is depicted in second row, the subtractive diffusion representation in second row, and the probabilistic edge detector in third row. The last row show the result obtained from the last representation.

References

[1] S. Alliney and C. Morandi, "Digital image registration using projections", IEEE Transactions on Pattern Analysis and Machine Intelligence., vol. 8, no. 2, pp. 222-233, 1986.

[2] M. Irani and P. Anandan, Robust Multi-Sensor Image Alignment, Proc. IEEE Internation Conference on Computer Vision (ICCV 1998), India, pp. 959-966, January 1998.

[3] S. Makrogiannis, I. Vanhamel, H. Sahli, S. Fotopoulos, Scale-Space Segmentation of Color Images Using Watersheds and Fuzzy Region Merging, Proc. Int. Conference on Image Processing (ICIP 2001), 7-10 October 2001, Thessaloniki, Greece, Vol. 1, pp. 734-737.

[4] E. Parzen, "On estimation of a probability density function and mode", Ann. Math. Stat. vol. 33, pp1065-1076, 1962.

[5] P. Perona, J. Malik, Scale-Space and Edge Detection Using Anisotropic Diffusion, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 7, pp. 629-639 (1990).

TASK-3: TRACKING TARGETS AND SPN FORMATION OF PATTERNS

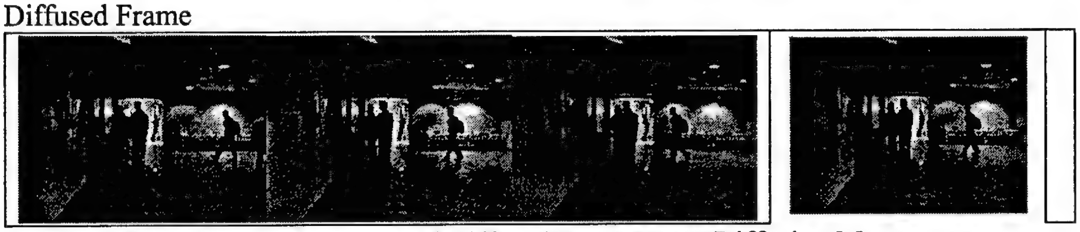
1. DETECTING AND TRACKING MOVING OBJECT IN SEQUENCES OF IMAGES

In the previous effort (Sept.2004) we have generated the first results from the merging of the Pixel Flow Functions (PFF) method with the Local Global Graph (LGG) method for detecting changes and patterns in sequences of images. Some problems were reported with the sensitivity of the PFF method. Here, most of these problems have been resolved using pre-filtering and the LGG method with segmentation, and results are presented below. Also, the next step is to detect and extract the target patterns or differences and define their formation.

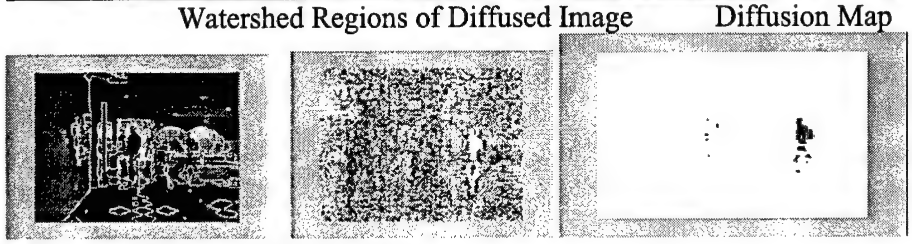
1.1. Spatio-Temporal Diffusion on Real-Tracking Changes in Sequences

First Frame Second Frame Third Frame

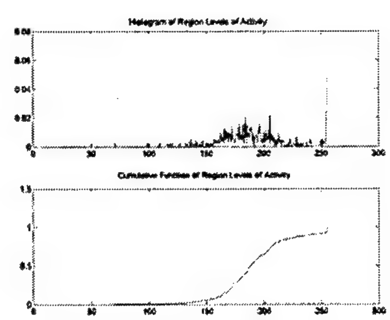
Diffused Frame



Watershed Regions of Diffused Image Diffusion Map



Regions of High Level of Activity



PFF's are strongly influenced by noise and coding artifacts, when employed for detecting differences without any pre-filtering process. On the other hand, PFF's are very efficient for tracking the previously recognized and segmented objects.

1.2. Aerial Pictures with PFF Method and Segmentation

0. Original Images

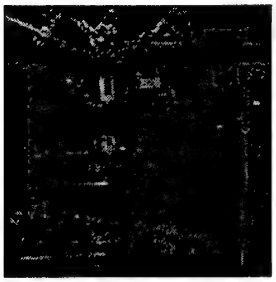
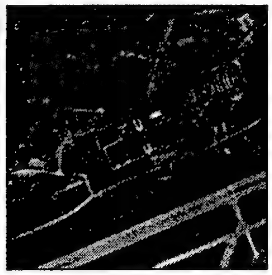
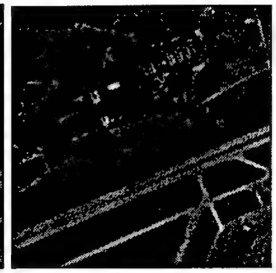




Image 1 Image 2 Image 3

1. Non Linear Pre-filtering

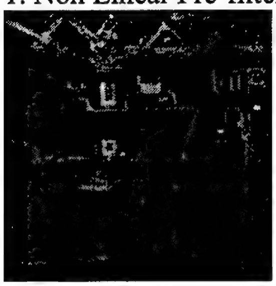
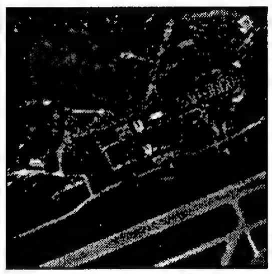
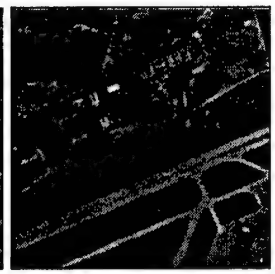




Image 1 Image 2 Image 3

1.3. Intensity Histogram

Image 1

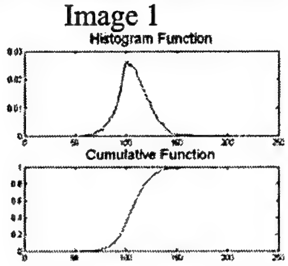


Image 2

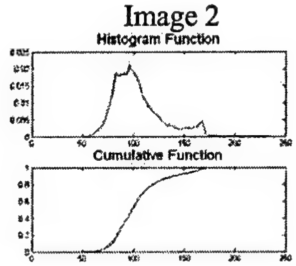
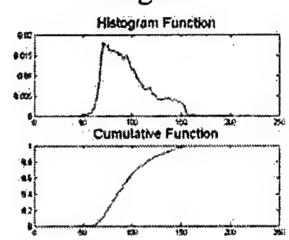


Image 3



1.4. Calculation of Gradient Magnitude and Orientation

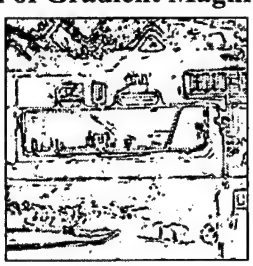
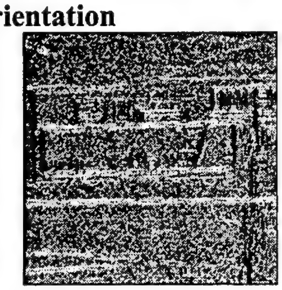



Image 1

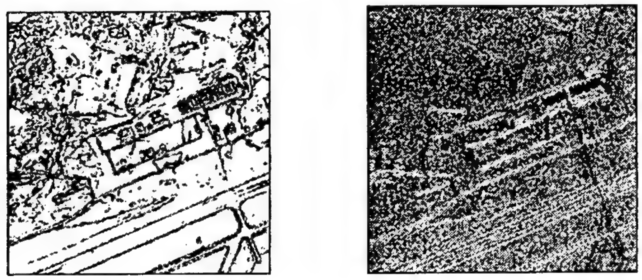


Image 2

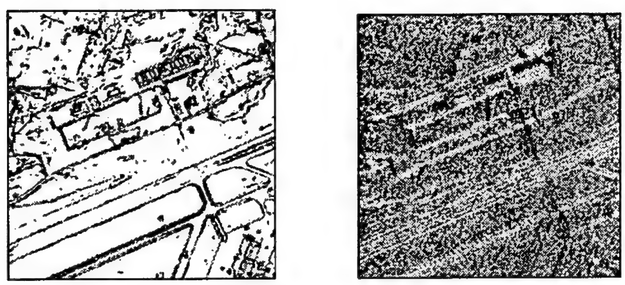
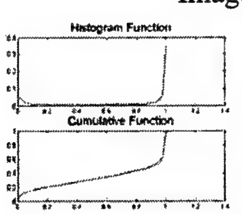
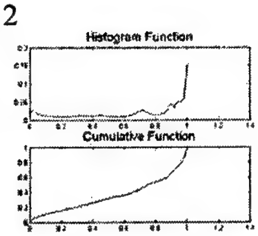


Image 3

Image 2





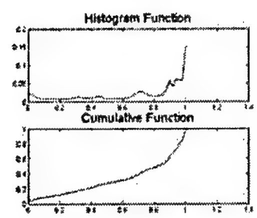


Image 1 Image 2 Image 3

In the above histograms it is obvious that the horizontal orientation is dominant in these three images (maximum probability for cos(a)=1=>a=0). A second conclusion is that the second and third images we have a remarkable percentage of pixels with diagonal orientation cos(a)=0.7=>a~45 degrees). In the first image this does not hold. In the first image it is obvious that the edges have either horizontal cos(a)=1, or vertical coas(a)=0 orientation.

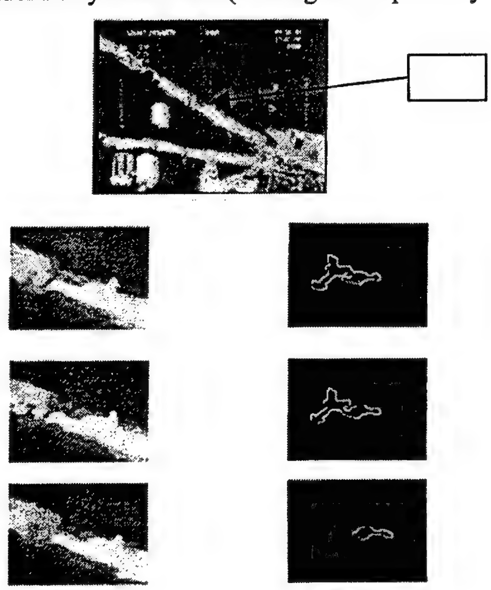
Another Option is to attempt to detect the shadows of the buildings which have low intensity values.

The application of a thresholding operation produces the following result below. We can see that the apart from the building shadows some other irrelevant dark areas have been detected too. Similar results are produced when a double thresholding operation is applied.

2. TARGETS TRACKING IS SPN FORMATIONS

Motion detection, recognition and tracking are also challenging problems[8]. These problems are associated with the representation of the objects to be detected, recognized and tracked. A good representation of an object offers a better detection and recognition performance. In addition, *a priori* knowledge of the 3D object views contributes to an accurate detection and recognition-tracking. Some techniques use perceptual constraints among various 3D primitives in space in order to group them and reconstruct the underlying surfaces [1,2]. Some other techniques use probabilistic approaches for detecting and tracking multiple objects [3,4] and others non-probabilistic ones [5]. The methods above present some difficulties in distinguishing various objects when they come close to each other. Some methods with a good performance are based on the multiple hypothesis tracking algorithm that provide a Bayesian framework for motion analysis of multiple objects[6-8]. These methods offer the advantage of handling statistical data associated with the initiation, termination and assigning measurements to tracking. In addition, most of hem make use of the Hausdorff algorithm, which employs the model based matching that preserves the shape and view point information of the objects offering a more robust tracking [7,8]. The methods mentioned above offer very good results in case the detection-tracking of motion is real in the navigation environment and not a projection on a wall, like a movie. In our case we intend to use range sensors images for accurately detecting real motion. In addition, for the recognition part we make use our 3D recognition model, which is based only on six views of an object. In other words it synthesizes others views from the models of those six [10,11]. In order to make our methodology robust, we plan to employ both an image and range based motion detection and tracking techniques.

This is an example of a moving target (tank) on a road with trees. Our method was able to detect track and extract the target under noisy conditions (the target was partially covered by bushes and a tree in the last frame).



Steps of the Objects Tracking Algorithm.

A. A spatio-temporal anisotropic diffusion method is first applied that uses the information of the current, previous and next frames in the sequence. This method is based on the anisotropic diffusion theory [PeronaMalik] by inserting a temporal variable in the heat diffusion equation. This smoothes out adaptively the areas of spatial and temporal homogeneity.

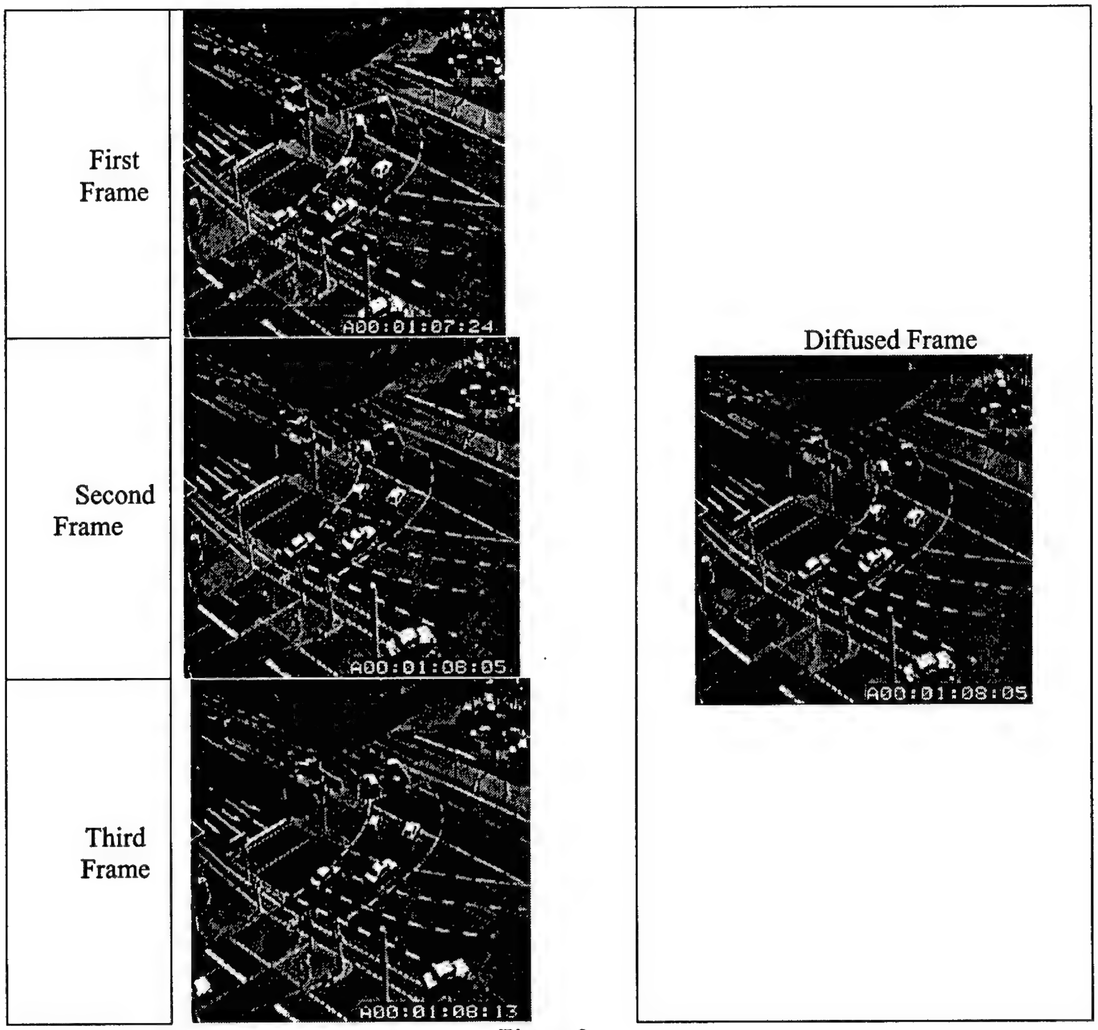
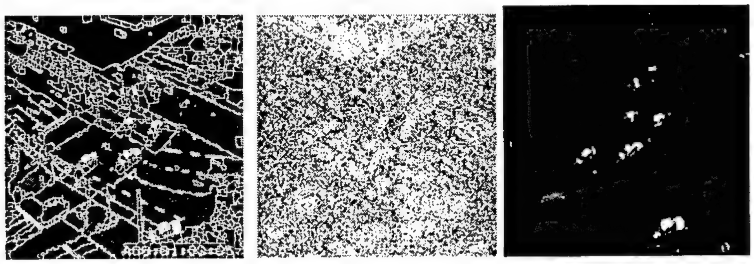


Figure 2.

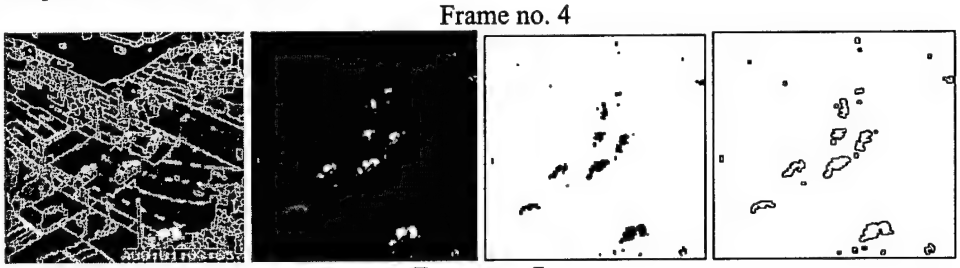
B. A watershed based segmentation algorithm is applied on the diffused image. This will produce higher segmentation detail in the areas that are not spatio-temporal homogeneous, see below.



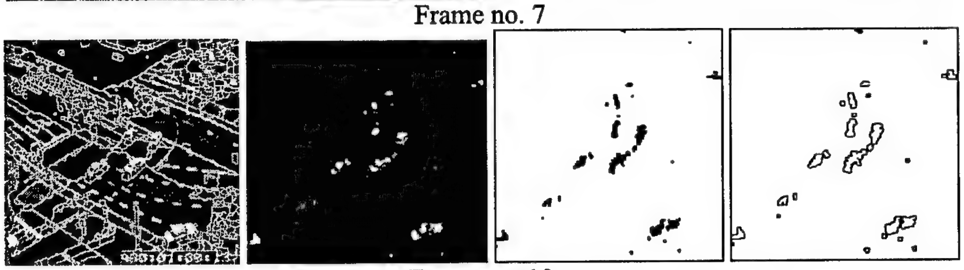
C. The Displaced FrameDifference is estimated between the Diffused frame and the current frame, see above.

D. A process to identify the active regions follows. First, the level of activity is calculated as the ratio of the sum of dfd pixels in a watershed region over it's area. A thresholding operation follows to detect the most active areas in the frame which are also decomposed into watershed regions. Areas of high activity divided into watershed regions, see above the last picture. Below are examples from different fames:

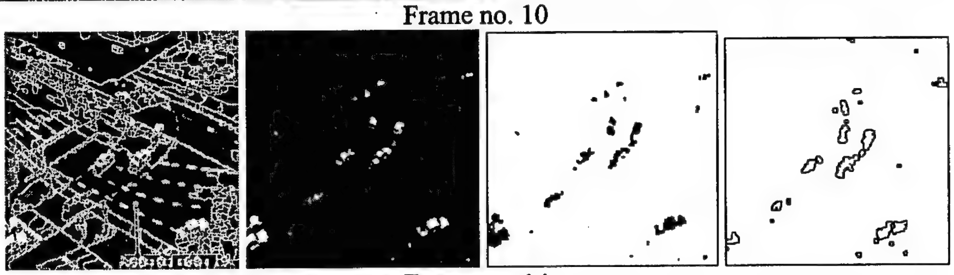
Frame no. 4



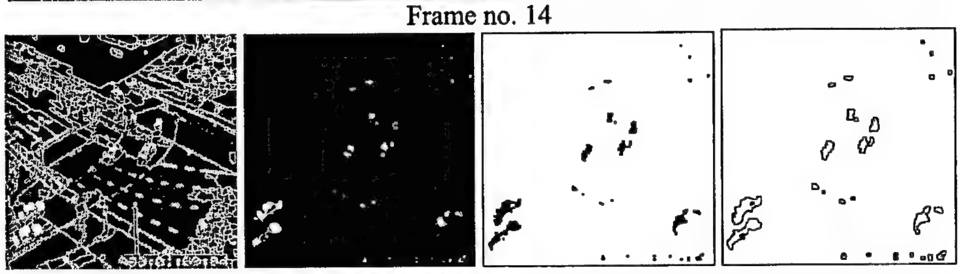
Frame no. 7



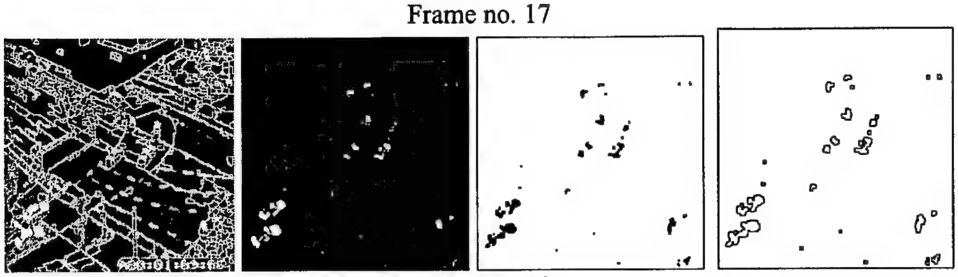
Frame no. 10



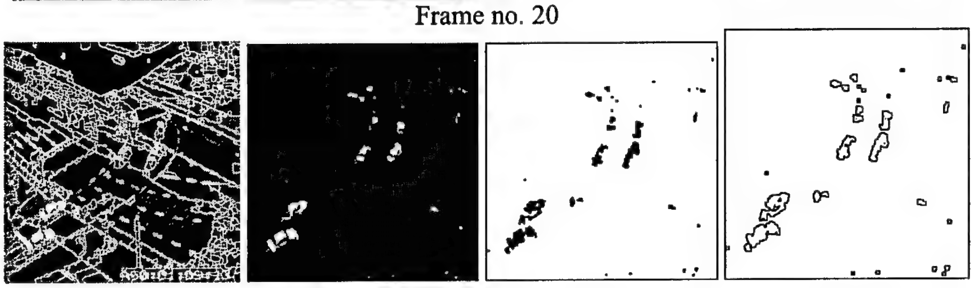
Frame no. 14



Frame no. 17



Frame no. 20



SPNG Associations

The SPN graph Model

A Petri-net model is a more than 40 years old methodology developed by Petri. Since then, thousands of publications and numerous of variations and applications have been presented around the globe. An SPN is graph of an object that has k different states (Places P_i, i=1,2,..k). Each place P_i has its own structural features transferred from the corresponding graph node N_i. The transitions t_{ij} and t_{kl} represent relationships among the same parts of a target and a stochastic distribution of time required to fire that transition. Here we make use of the stochastic Petri-net (SPN) model in a form of a graph and we take the advantage of the SPN properties (timing, parallelism, concurrency, synchronization of events) for our synergistic methodology [8,10,11].

Here we presents the results of SPN graphs associations for detecting formations from moving targets or patterns, see figures 3,4,5. More specifically, when the changes are detected and tracked in different frames, the Local Global (L-G) graph method is used to establish a local graph for each region. These region-graphs then associated for developing the global graph that associated all the region-graphs. This is the association pattern that represents the formation. By tracking these formations we have better understanding of the changes that take place in sequences of frames.

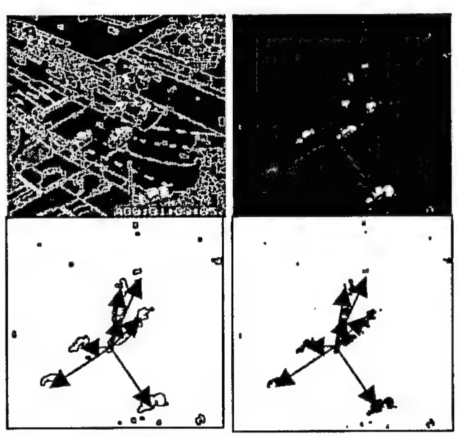


Figure 3: It shows the SPN graph associations for formations (frame No-4)

Frame no. 4 Frame no.7 Frame no. 10 Frame no. 14

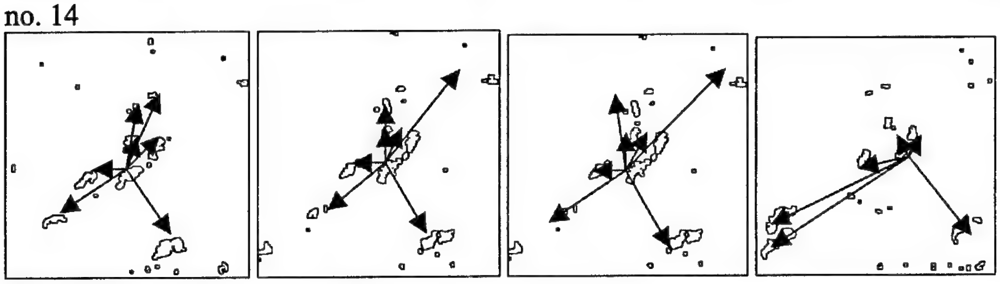


Figure 4: Formations taken from different moving patterns of targets in different frames

Tracking Patterns of Formations

In this sub-section we present the SPN graph formations. In particular, in each frame the changes based on motion are detected and extracted and their shapes are isolated from the background image. Then, these shapes, that may represent moving targets or objects, are described by using Local graphs and their relative locations in the frame is associated with Global graphs. This means that these changes (objects or targets) are fully represented by the L-G graphs. At this point, we take these L-G graphs representations (or formations) from each frame and we again associate them with SPN graphs in order to explain (or represent) their transitions from one frame to the next (or from one state to the next). As it is known well SPN is capable for representing state transitions efficiently [1,2,12,13]. Figure 5 shows four frames, their L-G graphs formations and the associations of these formations into global formations that show the transition (flow) of each change (or target) from frame to frame. Colors are used to illustrate the transitions and flow using token, which represent the cause of these transitions. In this particular example the token are associated with traffic rules. The important conclusion here is that these formations and their associations can be used to predict the behavior of the patterns of formations.

In the example presented here, the traffic rules provide the necessary information that assist us to project the new formations of these changes.

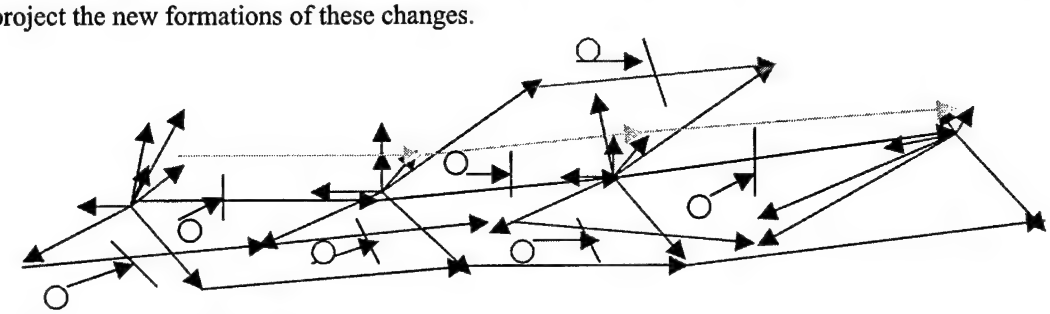


Figure 5: Tracking patterns of Formations in different frames. Due to limitations of colors only 4 pattern formations are illustrated. The SPN graph represents the transition from one state to the next. The circles represent the tokens that activate the transition. Due to the complexity of the diagram only a few tokens and transitions are presented in this picture.

References

[1] G.Guy and G.Medioni, Inference of surfaces 3D curves and junctions from sparse noisy 3D data, IEEE T-PAMI, 19,11, 1997, 1256-77.

[2] S.Kaneko, et.al. Robust matching of 3D contours using iterative closest point algorithm improved by M-estimation, Pattern Recognition Journal, 36, 2003, 2041-47.

[3] J.MacCormick and A.Blake, A probabilistic exclusion principle for tracking multiple objects, IEEE Conf. on CVPR, 1999, Greece, 572-578
[4] C.Rasmussen and G.Hager, Probabilistic data association methods for tracking complex visual objects, IEEE T-PAMI, 23,6,2001, 560-76
[5] I.Haritaoglu, et. al. "Real-time surveillance of people and their activities, IEEE T-Pami-22,8, 2000, pp.809-830.
[6] T.Cham and J.Rehg, A multiple hypothesis approach to figure tracking, CRL98/8 TR-1998
[122] S.Blackman et.al. Applications of multiple hypothesis tracking to shipboard first tracking, SPIE 2759, 441-452, 1996
[7]E.Polat, et. al. A 2D/3D model based object tracking framework, Pattern Recognition Journal, 36,2003, pp. 2127-2141
[8] N. Bourbakis, A Neural-based KB using SPNGs in sequence of images, AIRFORCE & SUNY-B-TR-1991,45 pages, Nov. 1991.
[9] J.Gattiker& N.Bourbakis Representation of Structural and Functional Knowledge using SPN Graphs, Proc. IEEE Conf. on SEKE 1995, MD.
[10] N.Bourbakis, J.Gattiker and G.Bebis,A model for interpreting human activities and events from video, IEEE Conf. on TAI-2000, Vancuver, Canada, pp.132-139.
[11] N.Bourbakis, EIRMA: a formal language for representation and interpretation of activities and events based on LG graphs, IEEE SMC, sub.
[12] T.Murata, *Petri nets: properties, analysis and applications*, Proc. of the IEEE,7,4,1989
[13] T.Murata, V.Subrahmanian and T.Wakayama, *A Petri net model for reasoning in the presence of inconsistency*, IEEE T-KDE, 3,3,1991

TASK-4: RECOGNIZING PATTERNS OR OBJECTS

1. Introduction

For image understanding and object recognition, many methods are proposed. The robustness of a recognition system, however, is under a major challenge when the background is highly textured and the discrimination among included objects in the same scene is not well determined. Moreover, the lack of structural information of different objects with similar texture characteristics could be wrongly classified into the same category. In [8,9] the authors consider the object spatial relationships as the measure of image similarity. In [7,10,29] a closed-form representation for a model and an object is used . Thus, the shape similarity is expressed as the amount ofthe model deformation energy needed to align two shapes. This method can match an object with deformation other than rigid transformation. But it assumes that the shape has been segmented from the background, and the mathematic shape representation is sensitive to some kinds of deformations, for example, a cut in a ring.

Wavelets and multi-scale methods are proposed to match 2-D shapes [4,11,14,17,38]. Those methods use coarse-to-fine representations to match 2-D shapes. Since they consider an object as one solid region, those methods are not appropriate to multi-region object recognition. The shape is also represented by its border points or by primitives, such as lines or splines [14,16] and then the object's shape similarity problem becomes a point sets similarity problem [1,5,12]. Huttenlocher *et al* uses an efficient algorithm to compute Hausdorff distance between two point sets to recognize an object [1]. The similarity function could be a post-probability function or Hausdorff distance. These methods only compare and recognize single shapes. If the object is composed of several parts, they cannot perform recognition since no object structure is considered by these methods. Generally, they require relatively clear and accurate object shapes, i.e., they skip the segmentation step and assume that an accurate shape is available. It is not always realistic.

One of the most widely used and quoted curve fitting approaches was presented by Pavlidis [26]. It sets a maximum deviate threshold as a fitting criteria. Any point that falls in this error bound can be fitted by a current line segment. However, it is too simple to give satisfying fitting results, especially when the interested region has both a long edge and rich details. In [34] the authors proposed a multi-primitive fitting method. The breakpoints of the curve are divided into corner and smooth joints. Fitting is done between consecutive breakpoints and it is threshold free. However, for a curve with noise, this method will create erroneous breakpoints and degraded fitting performance. In [19] the authors used circular arcs and ellipses respectively to represent curves. The methods are good when the object has circular or ellipse shape. Otherwise, they do not show much fitting performance improvement but have high computation complexity.

The graph method is used to apply spatial constraints to key nodes. There are many ways to represent graphs, such as Voronoi Tessellation and Delaunay Graph [19-22]. For the recognition of a pattern, or a model different graph-based techniques have been proposed. In particular, a graph editing method is used to compute graph distance and, in turn, the graph distance becomes the measure of image similarity [2,13]. Sub-graph isomorphism techniques are employed for perfect matching of one graph part with another graph [15].

A graphical template is also proposed to generalize the graph registration problem [3,5]. In [3] the authors use decomposable sub-graphs of the template graph to find the optimal match to a subset of the candidate point set. In [5] also the authors use the Dual-Step EM registration algorithm to solve the point-correspondence match problem. Generally speaking, the point registration issue is the bottleneck of a graph matching process and currently there is no method that can solve all registration problems. In our method, by utilizing region information in each graph node, the complexity of finding point correspondence is greatly reduced [6,24,30-32].

Relational graphs are considered as a good approach to describe pictures or scenes for pattern recognition [2,18,24,26,27,28]. In [2] the authors used a relational graph to represent characters. They proposed descriptive graph grammars as rules to organize and compare graphs. In [28] the author used a relational distance measurement for model-based matching . There are, however, some common limitations associated with graph matching problem. First, the matching of the model to the data image is node location driven. In other words the matching criterion is minimizing the mean square over all pixels of the difference between the model and the data image. This does not ensure that specific points of interest or landmarks be matched with great precision. Secondly, because of the inherent non-linearity of the problem, and the fact that the deformations are highly dimensional, the computational tools for calculating the match must use relaxation techniques, which runs the risk of converging to a local minimum that corresponds to a poor match.

The proposed here *Local-Global* (*L-G*) graph method adds local part information into the graph [24]. The graph is a more accurate representation of an object. Thus, a non-linear graph matching function is avoided and by combining the *Fuzzy-like Reasoning Search* (FRS) method and the *L-G* graph method the object recognition accuracy without increasing computation complexity is improved. The robust recognition of objects in complex images is still an open scientific problem in the computer vision field, as mentioned above. In this paper the use of the L-G graphs assists the synthesis of segmented regions [06] for creating a desirable object with a maximum confidence against a database of known objects. In addition, a graph based incremental learning process takes place during the synthesis of regions to define new complex objects.

2. Region Contour Fitting and Local Region Graphs

Notation : G_M and G_D are the graphs of an object model and an image respectively. M and D are the graph node sets of an object model and an image. E_M and E_D are the graph edge sets of an object model and an image.
Notation: The edge set is bi-directional.

2.1 The Border Curve Fitting

After the application of an image segmentation method, a set of color regions are generated. The normalization process of the region's borders leads to an appropriate curve fitting approach, which will assist the generation of the region graphs and the synthesis of the neighboring regions.

In this section we present our approach to curve fitting. The key point of the curve fitting accuracy is how to select the deviate threshold [30]. It generates more fitting lines for large-scale images than small-scale images, even if they contain an identical scene. To fix this problem, a relative line fitting threshold is used here.

The relative error th_{rel} is a threshold proportional to a current line segment's length. th_{rel} is defined as

$$th_{rel} = \mathbf{min}(th_{max}, \mathbf{max}(Len * p, th_{min})) \qquad (1)$$

where *Len* is the length of current line segment, *p* is predetermined percentage, th_{max} and th_{min} are two extreme thresholds. For long segments, it tolerates big deviate error and for short segments it uses a small threshold to produce accurate fitting results. But it still cannot fit well at the end of a long line segment. Because of the large *Len*, it sets a large threshold th_{rel}. The big th_{rel} may merge pixels that follow the current line segment though they have a very different trend. Figure 1 shows this example.

The *fitting error gating* technique presented here uses a new fitting-error bound computing approach and can offer a reasonable solution to the problem. A similar approach for fitting straight-line segments with unevenness was proposed in [35,36].

The Fitting Error Gating (FEG) Technique

The FEG approach dynamically determines the fitting deviate bound at every fitting step based upon the current fitting error. In the fitting process, we try to use the minimum number of line

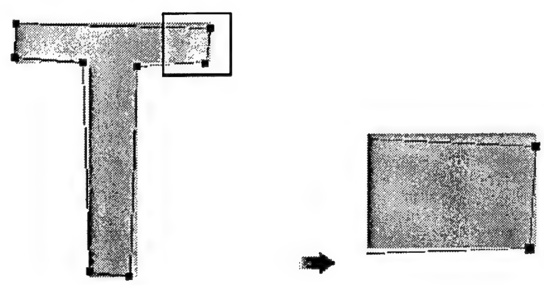


Figure 1: Relative threshold Curve fitting example

segments to fit the
border curve, i.e., at every step we try to fit the most possible points with one line. Here, a fitting line has to pass the start and the end points and fit, or it has to be as close as possible to the majority of the rest of the points remaining. This requirement ensures line connectivity, which is a prerequisite for the generation of the local region graph.

Thus, the main curve fitting process steps are:

1. *Set initial fitting Err.*
2. *Set the starting point pt_s as the first point and the ending point pt_e as the last point.*
3. *Compute the fitting line function y=Ln(x).*
4. *Compute the fitting deviate threshold th based on the Err.*
5. *Compute the new fitting error Err and the maximum deviate error Er_d.*
6. *if Er_d >th then $pt_e = (pt_s + pt_e)/2$; goto step 3*

else
  if (pt_s =first point && pt_e =last point) || (pt_e - pt_s <=1) then stop
  else $pt_s = (pt_s + pt_e)/2$; *goto step 3*

The curve fitting method works similarly to the known binary tree technique, because at every step the possible vertex (or stop point) range is narrowed down to half of current range. For a curve length of n (points), the maximum fitting steps are $[log(n)]$. The search process is illustrated in Figure 2 and the FEG in Figure 3.

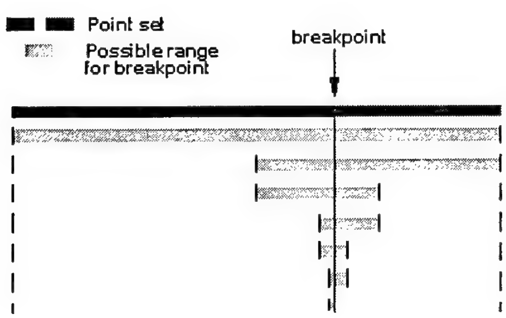


Figure 2: Binary fitting process illustration

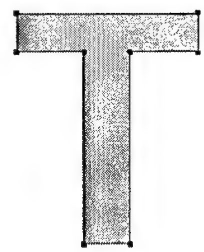
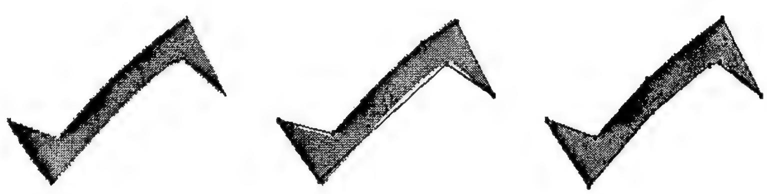


Figure 3: Fitting error gating method result

At this point, the current average fitting error as a gate factor *GT(n)is:*

$$GT(n) = \frac{1}{n}\sum_{i=1}^{n} e_i^2 \qquad (2)$$

where n is the total number of fitting points, and e_i is the distance from i_{th} point to the fitting line. Also, the maximum deviation error, Er_d, is computed at the current fitting step.



(a) Original object **(b) Fitted with relative threshold** **(c) Fitted with our method**

Figure 4: Curve fitting example

Thus, the dynamic error bound is computed by considering the gate factor *GT(n)*

$$th_{rel} = \mathbf{min}(th_{max}, SE(GT(n)), \mathbf{max}(Len * p, th_{min})) \qquad (3)$$

where the function *SE(x)* is defined as

$$SE(x) = \begin{cases} Er_d, & \text{if } x < th_{GT} \\ x \cdot Er_d / th_{GT}, & \text{if } x \geq th_{GT} \end{cases} \qquad (4)$$

where Er_d is the maximum deviate error of the current fitting, th_{GT} is the gate factor threshold. In equation (3), if $GT(n)$ is smaller than th_{GT}, it means all points in the current fitting point set are close to the fitting line (least square error). For the human's perception point of view, it is an accurate fitting of the current point set; in other words, any change of this fitting line segment is more likely to degrade the fitting performance, i.e., generate erroneous fitting results. In this case, $SE(\cdot)$ returns the current maximum deviate fitting error. Also, the maximum fitting threshold will be confined to no more than Er_d. If $GT(n)$ is bigger than th_{GT}, $SE(GT(n))$ returns a value proportional to $GT(n)$. For example, in figure 1, the borders of character T are perfect horizontal and vertical lines. The fitting error $GT(n)$ is 0. Thus, at the next fitting step, the fitting threshold is set to 0. Thus, only points with the same trend will be fitted by the current line. It excludes the points that belong to other lines. The fitting result is shown in figure 3. Thus, it has a relative error threshold characteristic, i.e., generates a long line segment in a low frequency area and a short line segment in high frequency area. In addition, with the application of the error gating technique, the fitting performance is improved. Below is another example. This method mentioned above offers a better accuracy versus original methods proposed in [19, 26, 34, 35].

2.2 The Local Region Graph [2, 26,31]

From the curve fitting result, we build the local graph of the current region. The border curve is represented by connected lines. Thus, the shape is expressed as

$$SH = \Upsilon \{Ln_j \cdot R^c_{j,j+1} \cdot Ln_{j+1}\}$$

$$= Ln_1 \cdot R^c_{1,2} \cdot Ln_2 \cdot R^c_{2,3} \cdot Ln_3 \cdot R^c_{3,4} \cdot \ldots Ln_{n-2} \cdot R^c_{n-2,n-1} \cdot Ln_{n-1} \cdot R^c_{n-1,n} \cdot Ln_n \qquad (5)$$

where n is the number of lines, $j \in [1, 2,\ldots, n\text{-}1]$, Ln_j and Ln_{j+1} are two consecutive curve lines, $R^c_{j,j+1}$ is the relationship between Ln_j and Ln_{j+1}.

The complete representation of a shape SH, however, requires the determination of two more factors.

i. the individual properties P_j of line Ln_j,

P_j = {*sp(starting point), l(length), d'(orientation), cu(curvature)* }

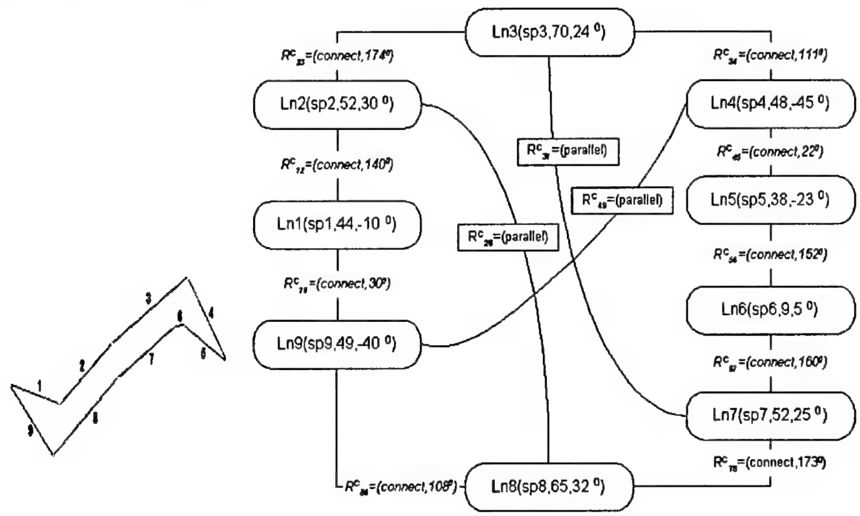
where the index j indicates the appropriate segment.

ii. The relationships RL_{ij} among the line segments

RL_{ij} = { ***c****(connectivity),* ***p****(parallelism),* ***rd****(relative distance),* ***rm****(relative magnitude),* ***sy****(symmertry)* }

where, the sub index ij means the relationship between line i and line j.

Thus, the line segments Ln, their properties P_i and the relationships R_{ij} among the segments are defined for a sufficient description of the current region shape. Figure 5 shows a sample region and its local graph with attributes.



(a) Line fitted object

(b) object's local graph

Figure 5: This is an example of local graph. Left image is the object or a single region. The number besides every line is the index. Its local graph representation is shown on right. For simplicity, only some fields of local graph are shown

3. Wavelets for Contour Matching

3.1 Single Region Matching

The starting point of the curve matching problem is to match a single region object. In particular, we suppose there are two closed curves $f(t)$ and $g(t)$ and assume $g(t)$ is the transformed counterpart of $f(t)$. The goal is to recover the parameters of a geometric transformation matrix that best maps a curve $g(t)$ to $f(t)$. We represent each point in the region point set in its homogeneous form.

3.1.1 General Geometry Transformation [37]

In order to assist the reader in understanding this concept we briefly present the geometry transformation here. We consider a closed 2D curve, $f(t)$, where t denotes a parameter. A 2D curve can be determined by its Cartesian coordinates of all points. Thus a curve is defined as

$$f(t) = \begin{bmatrix} x(t) \\ y(t) \\ 1 \end{bmatrix}, t=1,2,\ldots,m$$

where m is the total region point number, and $g(t)$ is the transformed form of $f(t)$.

Generally, the basic geometry transformation is composed of translation, rotation, and scaling. In the case of the affine transformation, which includes reflection and shearing, there are six free parameters. These model the two components of the translation of the origin on the image plane, the overall rotation of the coordinate system, and the global scale, together with the parameters of shearing and reflection. These parameters can be combined into an augmented matrix that takes the form [37]

$$\phi = \begin{bmatrix} \phi_{1,1} & \phi_{1,2} & \phi_{1,3} \\ \phi_{2,1} & \phi_{2,2} & \phi_{2,3} \\ 0 & 0 & 1 \end{bmatrix}$$

The solution for six parameters simultaneously is very difficult. In order to simplify this problem, we assume there is no shearing and reflection transformation (in fact the reflection transformation is considered later). Thus, the transform matrix with translation, rotation, and scaling, can be expressed as

$$\phi = \begin{bmatrix} rs_{xx} & rs_{xy} & trs_{xx} \\ rs_{yx} & rs_{yx} & trs_{yy} \\ 0 & 0 & 1 \end{bmatrix}$$

and the transformed curve ***g(t)*** is,

$$g(t) = S \cdot M_\theta \cdot f(t) + T$$

$$\begin{bmatrix} x'(t) \\ y'(t) \\ 1 \end{bmatrix} = \begin{bmatrix} rs_{xx} & rs_{xy} & trs_{xx} \\ rs_{yx} & rs_{yx} & trs_{yy} \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x(t) \\ y(t) \\ 1 \end{bmatrix} \tag{6}$$

The four elements $\underline{rs_{ij}}$, where ij take values x and y, are the multiplicative rotation-scaling terms in the transformation that involve only rotation angles and scaling factors. Elements $\underline{trs_{xx}}$ and $\underline{trs_{yy}}$ are the translation terms containing combinations of translation distances, pivot-point and fixed-point coordinates, and rotation angles and scaling parameters.
The scaling process can have different scale factors for the x and y coordinates, s_x, s_y. It deforms object contours with a different ratio in the x and y directions. Note that most of the imaging elements have the same scale factor in both x and y directions. Thus, we set the scale factors for both x and y directions the same, i.e.

$$s \equiv s_x \equiv s_y.$$

The scaling matrix $\boldsymbol{S}$ is reduced to s, that is

$$g(t) = s \cdot M_\theta \cdot f(t) + T \tag{7}$$

3.1.2 Translation parameter

The translation matrix $\boldsymbol{T}$ has two elements t_x and t_y. Let $\boldsymbol{F_{cen}}(f(\cdot))$ be the function which computes the centroid of curve $f(\cdot)$. In the geometry transformation, rotation and scaling don't change the centroid location (x_0, y_0). Thus,

$$\begin{aligned} F_{cen}(g(t)) &= F_{cen}(s \cdot M_\theta \cdot f(t) + T) \\ &= F_{cen}(s \cdot M_\theta \cdot f(t)) + F_{cen}(T) \\ &= F_{cen}(f(t)) + T \\ \rightarrow \quad T &= F_{cen}(g(t)) - F_{cen}(f(t)) \end{aligned} \tag{8}$$

Since the centroids of the two curves are known, we move their curves centroids to the origin point. It will simplify the following process. Figure 6 shows an example .

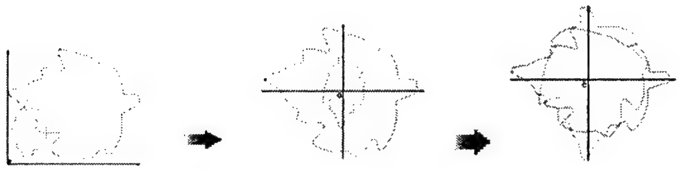


Figure 6: Translate parameter example   Figure 7: Scale parameter example

3.1.3 Scale parameter

To find the scale parameter the momentum is chosen. The momentum is a measure of an object's mass distribution. It is defined as [38]

$$Mom = \frac{1}{N}\sum_{i=1}^{N} m_i \cdot \left\| p_i - p_{centroid} \right\|^2 \tag{9}$$

where N is the number of curve points, m_i is the mass weight at point p_i, $p_{centroid}$ is the centroid point of the closed curve. Finally, $\|\bullet\|$ is a kind of norm, such as a Euclidean norm.
If the object's mass is evenly distributed, m_i becomes a constant value. Then, the momentum *Mom* is determined merely by its point distribution, i.e., determined by its shape. Thus, *Mom* becomes an object geometry shape description [25].
Translation and rotation do not change the shape of an object, so *Mom* is identical before and after translation and rotation. Thus, the momentum *Mom'* after scaling is,

$$\begin{aligned} Mom' &= \frac{1}{N}\sum_{i=1}^{N} m_i \cdot \left\| p'_i - p'_{centroid} \right\|^2 \\ &= \frac{1}{N}\sum_{i=1}^{N} m_i \cdot \left\| s \cdot p_i - s \cdot p_{centroid} \right\|^2 \\ &= s^2 \cdot \frac{1}{N}\sum_{i=1}^{N} m_i \cdot \left\| p_i - p_{centroid} \right\|^2 \\ &= s^2 \cdot Mom \end{aligned} \tag{10}$$

The scale process affects the curve parameter linearly, but the momentum ratio is proportional to the square of the scale factor, s^2.
Thus

$$s = \left(\frac{Mom'}{Mom} \right)^{1/2} \tag{11}$$

Figure 7 shows an example of adjusting scale parameter,

3.1.4 Rotation parameter

We suppose that the curve ***f'(t)*** is generated by rotation of the curve ***f(t)*** by angle Φ. The rotation is obtained on ***f(t)***'s centroid. The rotation matrix ***M*** is defined as [37]

$$M = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}$$

the rotated curve ***f'(t)*** is computed by

$$f'(t)=\begin{bmatrix}x(t)'\\ y(t)'\\ 1\end{bmatrix}=M*f(t)=\begin{bmatrix}\cos\phi & -\sin\phi & 0\\ \sin\phi & \cos\phi & 0\\ 0 & 0 & 1\end{bmatrix}*\begin{bmatrix}x(t)\\ y(t)\\ 1\end{bmatrix} \tag{12}$$

The problem associated with the equation (12) is that we do not know the point correspondence yet. When the rotation angle Φ is computed with the equation above, if $x(t)$, $y(t)$ will be substituted with a point on curve ***f(t)***, the correspondent rotated point on ***f'(t)*** is unavailable. In order to solve equation (12), the right point mapping order has to found.

3.2 Wavelet Coefficient of Border [4, 37]

Because of the advantage of the multi-resolution analysis ability, the wavelet technique is used to solve the rotation matching problem. A wavelet is defined as

$$\psi_{a,b}(t)=\frac{1}{\sqrt{a}}w(\frac{t-b}{a}) \tag{13}$$

where $w(t)$ is the initial wavelet, a is a scale coefficient and b is a shift coefficient.
One good property of wavelets is that its integral over all t equals to zero,

$$\int\psi_{a,b}(t)dt=0 \tag{14}$$

The computation of the ***f(t)***'s wavelet transform produces its wavelet transform coefficient $c_{a,b}$

$$c_{a,b}=\int f(t)\cdot\psi_{a,b}(t)dt \tag{15}$$

Under rigid motion and the affine transform, the curve ***f(t)*** can be shifted, rotated and scaled. But its shape doesn't change under this kind of transform. We can substitute ***f(t)*** in (15) with its transformed version [4]. Let ***f'(t)*** be the transformed curve. Thus

$$f'(t)=s\cdot M\cdot f(t+t_0)+T \tag{16}$$

where s is the scale coefficient, $\boldsymbol{M}$ is 2x2 rotate matrix and $\boldsymbol{T}$ is 2x1 translate vector.
Our interest is to find correlation between ***f(t)*** and ***f'(t)***. Let's exploit the ***f'(t)*** wavelet transform coefficient. Here, we use the same wavelet basis $\psi_{a,b}(t)$

$$\begin{aligned}c'_{a,b}&=\int f'(t)\cdot\psi_{a,b}(t)dt\\ &=s\cdot M\int f(t+t_0)\,\psi_{a,b}(t)dt+T\int\psi_{a,b}(t)dt\end{aligned} \tag{17}$$

from (14), we can simplify (17) as

$$\begin{aligned}c'_{a,b}&=s\cdot M\int f(t+t_0)\cdot\psi_{a,b}(t)dt\\ &=s\cdot M\int f(t+t_0)\cdot w(\frac{t-b}{a})dt\\ &=s\cdot M\cdot c_{a,b+t_0}\end{aligned} \tag{18}$$

Thus, we know that a curve can be reconstructed by its wavelet transform coefficient; hence, the transformed curve can be generated using the transformed wavelet coefficients with the same wavelet bases. By this way, we bypass the tough point-to-point correspondent task.
The Equation (18) shows if we can solve s, $\boldsymbol{M}$ and t_0, the matching problem will simplify to wavelet transform coefficient matching. Since the scale factor s and translate matrix $\boldsymbol{T}$ have been recovered, the only two variables here are the rotation matrix $\boldsymbol{M}$ and shift parameter t_0.
The rotation matrix $\boldsymbol{M}$ is determined by the rotation angle θ, so we define the wavelet coefficient error function $E_w(\theta,t_0)$ as

$$E_w(\theta,t_0) = \| c'_{a,b} - c_{a,b} \| \\ = \| s \cdot M \cdot c_{a,b+t_0} - c_{a.b} \| \quad (19)$$

Notice here that the scale factor s is known.

3.3 Univariate Search [38]

If we can find the minimum of error function $E_w(\theta, t_0)$, then the value θ and t_0 at the minimum error function value are the transformation parameters we need. Because of the increased computational complexity associated with changing all variables simultaneously, it is difficult to find the minimum value by computing all possible variable values. Practically, we have to consider techniques to reduce the total amount of computation. One solution is the *univariate search* method . In order to compute the minimum value of the multi-variable function $Y = f(x_1, x_2, ..., x_n)$ a feasible way is to change the variable one by one. Suppose that the variables are changed in their natural order, i.e., $x_1, x_2, ..., x_n$ (if this is not desired, they can always be re-numbered). The guiding idea behind *univariate search* is to change one variable at a time, thus, the function is minimized in each of the coordinate directions. The search process is graphically illustrated in Figure 8.

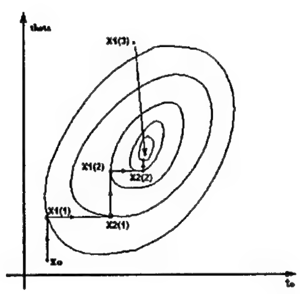
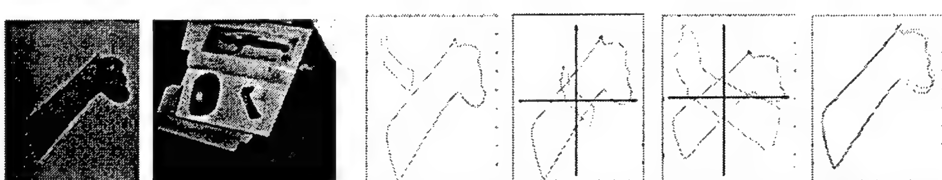


Figure 8: Univariate Search



(a) (b)(c) (d) (e) (f)

Figure 9: This shows a complete example of single region matching. (a) Model region (b) Object. (c) Original shapes. (d) After translation adjusted. (e) After scaling adjusted. (f) Final matching result.

Note that the SearchMinVal_$x1(t_0)$ and SearchMinVal_$x2(\theta)$ are search functions used to find the t_0 and θ to make the error function minimized along the directions x_1 and x_2.

Figure 9 shows an example of comparing single region using wavelets. It is also important to be mentioned that the local region graph is used as a criterion for determining the scaling threshold of the wavelet.

4. Matching Multiple Regions with the *L-G* Graph Method

For an object composed of more than one regions, the shape of every region cannot ensure two scenes are similar. If one scene has the same regions as another scene but arranged with different

relationship, these two scenes are totally different no matter how similar every region pair is. Here, the spatial relationships between corresponding regions represent an important constraint to the matching process. The location of spatial features serves as a natural choice (as landmarks) in relating multiple views of real world scenes. Differences in images of the same scene may be induced by the relative motion of the camera, or different illumination and the scene itself.

Rather than using the shape constraints to establish similarity correspondence, we use the constraints provided by the spatial adjacency of the regions. These constraints are relaxed by separately triangulating the data and model regions [23]. We use the neighborhood consistency of the correspondences in the triangulations to weight the contributions to the similarity function. In part of this section, we describe how the relational consistency is used in the matching process. In particular, we abstract the representation of correspondences using a bipartite graph. Because of its well-documented robustness to noise and change of viewpoint [21-23], we use the Voronoi Tessellation method, along with the Delaunay triangulation and the Local Global graphs as our basic representation of the image structure.

4.1 Voronoi Tessellation and Delaunay Graph

The dot patterns corresponding to the local feature positions of objects may prove to be relatively insensitive to 2-dimensional geometry transformation [23]. In addition to the transformation, the objects of the image scene may be noisy. The matching problem can be stated as follows: given two dot patterns, we want to know if one is a rotated, translated, and scaled version of the other. By using Voronoi tessellation and Delaunay graph, the pattern-matching problem becomes a two dot-patterns match problem. Given two dot patterns, one is called the Model pattern, and another, called the Object pattern, which is a rotated, translated, and scaled version of the model pattern. In [27] the authors consider matching with respect to translation, allowing perturbation of a point by at most a given threshold t. They attempt all possible translations that map a pair of points in one pattern onto a pair in another pattern, within the given tolerance t. In [39] the author compares the minimal spanning trees of the patterns in order to determine their degree of match. He attempts a matching between points in the two patterns with respect to the degree of the minimal spanning tree at the points, angles formed by the lines joining the points to their neighbors, etc. good matches are sued to establish correspondences between points in the two patterns. In [41] the author matches patterns by comparing ordered lists of boundary cells. This should have the effect of aligning the borders of the two patterns, thereby suggesting the potentially matching point pairs in the interiors of the patterns.

One limitation of the above methods is that they use the point position but do not use any information from the image or the image-region. Thus, basically they consider only the dot or point geometry relationship. Moreover, another limitation is that no point correspondence information is available in the dot pattern. They either need to permute all possible combinations, or use border shape to find the point correspondence.

Recently, in [38] the authors proposed a method that uses a dual-step matching method. The matching process alternates between estimating transformation parameters and refining correspondence matches, i.e., point registration. This approach recovers transformation parameters very efficiently. Nevertheless, in order to obtain a good result, the key points in the object have to be defined prior to match.

Suppose that we are given a set S of three or more points in the Euclidean plane. Assume that these points are not all collinear and that no four points are co-circular. Consider an arbitrary pair of points P and Q. The bisector of the line joining P and Q is the locus of points equidistant from both P and Q and divides the plane into two halves. The half plane H_P/H_Q is the locus of points closer to $P(Q)$ than to $Q(P)$. For any given point P, a set of such half planes is obtained for various choices of Q. The intersection $\mathrm{I}_{Q \in S, Q \neq P} H_P^Q$ defines a polygonal region consisting of

points closer to P than to any other point. Such a region is called the Voronoi polygon associated with the point [22]. An example of the Voronoi tessellation is shown in Figure 10.
The points, whose polygons share edges with the polygon containing a given point P are called P's
Voronoi neighbors.
The process of Delaunay triangulation generates relational graphs from the two sets of point-features. More formally, the point-sets are the nodes of a data graph

$$G_D = \{D, E_D\}$$

and a model graph

$$G_M = \{M, E_M\}$$

where D and M are the node sets of data (image) and model respectively. And the $E_D \subseteq D\times D$ and $E_M \subseteq M\times M$ are the edge-sets of the data and model graphs. The key to the matching process is that it uses the edge-structure of Delaunay graphs to constrain the correspondence matches between the two point-sets. This correspondence matching is denoted by the function $f\colon D \rightarrow M$ from the nodes of the data-graph to those of the model graph. According to this notation, the $f(i) = j$ indicates that there is a matching between the node $i \in D$ of the data-graph to the node $j \in M$ of the model-graph.
A Delaunay graph is robust to noise and geometry transformation. In another word, if the node set undergoes any kinds of transformations, the new Delaunay graph is the transformed version of the model Delaunay graph, with the same transform parameters. A simple example to show Delaunay Triangulation is invariant to translation, scale, and rotation, is shown in figure 11.

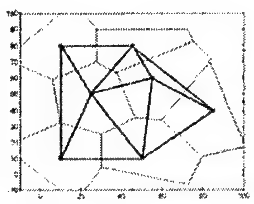


Figure 10: Voronoi Tessellation and Delaunay Graph

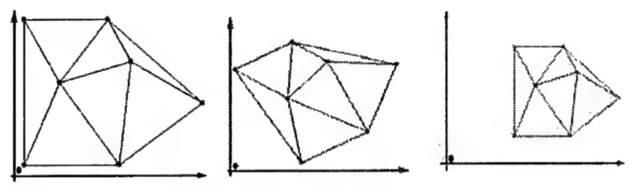


Figure 11: Delaunay graph of translated, rotated, and scaled point set

4.2 Region Node and Local –Global Graph

In the *L-G* graph scheme, the graph node is not a point but a region. Every *node* in the graph has not only point position information, but also all the characteristics of the region that it represents.
We define every node in the graph as

$$node = \{(x,y),\ color/texture,\ \mathbf{L},\ border,\ size\}$$

where

- *(x,y)* is the location of the node, which is identical to a correspondent region's centroid.
- *color* is the chromatic information of the region, the HSI color model is used.
  texture is a region's texture
- ***L*** is the local graph associated with this node (region).
- *border* is the object contour pixel set.
- *size* is the number of all pixels belonging to this region.

Texture is a well-researched property of image regions, and many texture descriptors have been proposed in the literature. Despite that, texture definition and representation remains an open research topic. We do not go deep here on the classical texture approach, but it is utilized in another work of ours. After introducing the local graph information into the global graph scheme,

the new method can handle both local (region) and global (object) information in the matching process, thus, we use the *L-G* graph to represent the new local-global graph method.

4.2.1 Represent image with *L-G* graph

Because the segmentation result is available here, we combine the *L-G* graph with the segmented region by means of the above definition. The fuzzy-like segmentation method divides the image into distinct regions. Every region is a characteristic of the image. The node has not only spatial location but also other region information, such as color, texture, etc. As previously defined, the object and model *L-G* graphs are:

$$G_D = \{D, E_D\},\ \ G_M = \{M, E_M\}$$

where D and M are node set; E_D and E_M are edge set. A node set is defined as

$$NS = \{node_i,\ i=1,2,\ldots,n\},\ n \text{ is the total node number}$$

Edge set E is a $n \times n$ matrix and defined as

$$E(i,j) = \begin{cases} 1, & \text{if } node_i \text{ connects with } node_j \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

Below is an example of a *L-G* graph, figure 12.

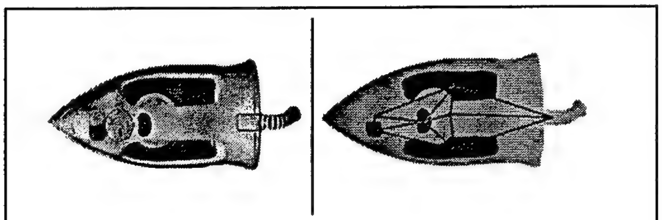


Figure 12: Sample image Ironer and its Delaunay Graph

5. Comparing Graphs

Generally, matching two graphs is in order to find the point correspondence between two graph node sets, which maximizes the likelihood between the two graphs, given the spatial constraint [3,5,30]. If the geometry position is only available, the matching process in general requires a permutation algorithm or a recursive method. In our scheme, the node correspondence can be solved by using the region color and the shape similarity, which can bypass the recursive or permutation method and reduce the computational complexity. Suppose the node correspondence has been established. Now the problem is how to compute the similarity between those two graphs, given the node correspondence, considering the geometry translation. Because the node correspondence is known, the graph similarity is determined by their relative spatial connectivity. In other words, the translation, rotation, and scaling do not change the graph spatial structure. The spatial structure of the graph is mainly represented by the angle, i.e., if the correspondent angles between arcs are similar, the graphs are similar too. We can prove that if the angles are matched to each other then two graphs' structures are also similar.

We define the angle similarity function $S_{ANGSIM}(\Delta\theta)$ by using two thresholds. One is θ_{th1}, the lower bound of angle difference, and the other is θ_{th2}, the upper bound of angle difference, see

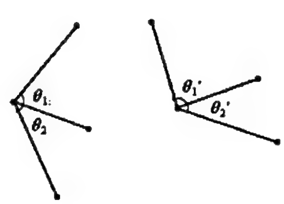


Figure 13: Compared nodes

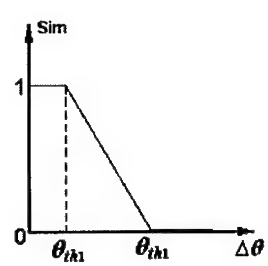


Figure 14: Angle Similar Function

figures 13 and 14. The similarity between correspondent angle is computed by the function $S_{ANGSIM}(\Delta\theta)$,

$$S_{\mathrm{ANGSIM}}(\Delta\theta) = \begin{cases} 1, & \Delta\theta < \theta_1 \\ \dfrac{\theta_2 - \Delta\theta}{\theta_2 - \theta_1}, & \theta_2 \geq \Delta\theta \geq \theta_1 \\ 0, & \Delta\theta > \theta_2 \end{cases} \tag{22}$$

The total similarity between two graphs Sim_G is

$$Sim_G = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} E(i,j)\cdot S_{\mathrm{ANGSIM}}(\theta_{i,j} - \theta_{i0}) \tag{23}$$

where N is the *node* number, $\theta_{i,j}$ is the angle of edge *(i,j)*, θ_{i0} is selected as the based angle for every node. The based angle θ_{i0} can be selected as the angle of the first arc associated with node *i*.

5.1. The *L-G* Graph Matching Scheme

The key idea behind the *L-G* graph is to use the local graph similarity as a constraint to the global graph. Thus the matching complexity can be reduced to an acceptable extent. This is the point where the *L-G* graph method surpasses other graph matching approaches.

Finding the node correspondence and the *PCRP* permutation

In our scenario, the node is a region. Here, every region is a meaningful part or characteristic of the interested object. We define that only regions with similar characteristics can be considered as a Potential Correspondent Region Pair (*PCRP*). Random matching of one node in the graph with nodes in another graph is not reasonably acceptable.

In the human perception system, color plays an important role in recognizing objects. It is reasonable to link node pairs that have similar color. The method proposed in this paper does not allow one node to correspond to nodes with very different color. For example, a red region, such as an apple, cannot correspond to a blue region, such as sky. The case of different colors is examined in [24,40]. Suppose a model set *M* has *N* nodes. A threshold th_c is chosen to filter out dissimilar nodes in the data graph. The process is:

1) *Initialize PCRP table, which has n entries, where n is the region count in the model graph.*
2) *For every region in the model graph, we compute color distances with all regions in the data graph. Any region with distance less than threshold th_c will be added into current PCRP table entry.*
3) *If none PCRP is found for a current region, an ERROR is returned.*
4) *Table Entry index increase 1 and go to step (2).*

Figure 12 shows an image, which has the ironer. The ironer model, a Delaunay graph is shown in figure 12 as well. Figure 15 shows the image's its segmented view. Figure 16 shows a complete example of selecting *PCRPs* and comparing the *L-G* Graph generated from the PCRP set.

After building all PCRP graphs, they are compared with the model graph shown in Figure 12. The equation (23) is used to compute the similarity value between the PCRP graph and the model graph. All the results are shown in Table 1. Figure 16 shows all PCRPs in the image. Remember, the PCRPs are selected only based on their color similarity. We have not applied any geometry and relationship constraints yet. The combinations of all PCRPs have 16 possible graphs. Figure 17 shows all the *PCRP* graphs.

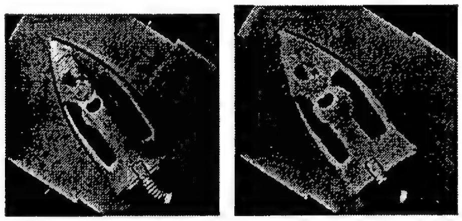


Figure 15: Data Image. Left is the original image and right is segmented by FRG method.

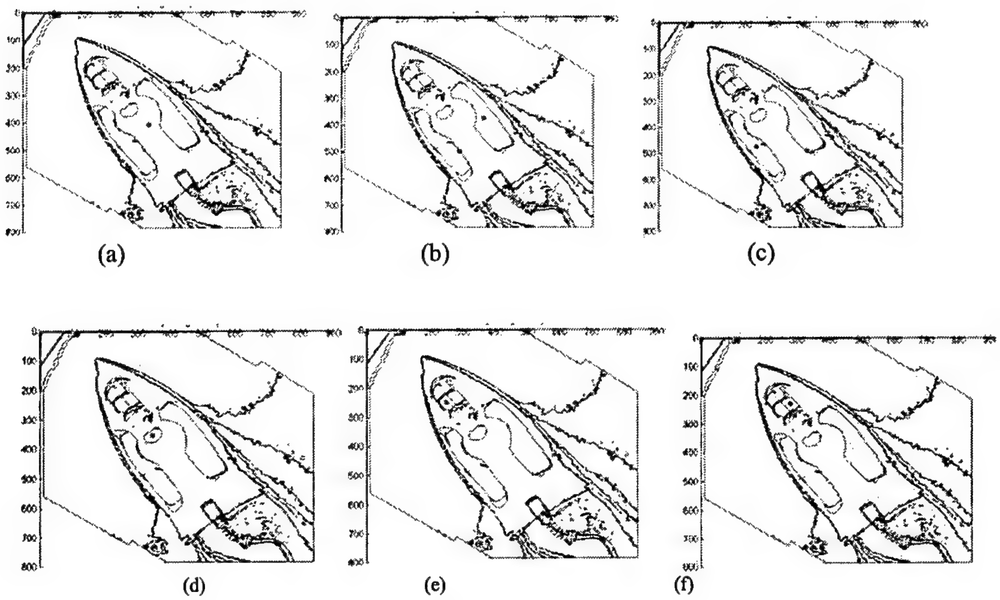


Figure 16 PCRP selection of every region. (a), (b), (c), (d), (e) and (f) correspondent to a region in the model, respectively

Table 1: Graph comparison result. All possible graphs from PCRP are compared with a model graph. There are 16 PCRP graphs and their similarity measure to a model graph are listed in the table.

Graph index	1	2	3	4	5	6	7	8
Graph SimVal	*0.98*	*0.83*	*0.46*	*0.38*	*0.13*	*0.27*	*0.18*	*0.16*
Graph index	9	10	11	12	13	14	15	16
Graph SimVal	*0.35*	*0.31*	*0.21*	*0.23*	*0.32*	*0.27*	*0.33*	*0.19*

(1) (2) (3)

(4) (5) (6)

(7) (8) (9)

(10) (11) (12)

(13) (14) (15)

(16)

Figure 17: 16 PCRP graphs from the examined image

5.2 The *L-G* Graph Relationship Checking

The example shown above indicates that only graph spatial constraints cannot ensure the finding of the right answer. In that example, two graphs are selected based on their graph geometry similarity. One graph is the right match but another is not. In an extreme case, it may have no right match in these selected graphs. Thus, we need to examine the validity further.

One factor we add to equation (23) is the shape similarity previously computed. The graph similarity is measured by every node similarity. In equation (23), we assume every node pair has the same weight. Nevertheless by considering the shape similarity of the two regions associated with the node pair, a large weight value is given to a node pair with a high shape similarity, and a small weight is given to the node pair with a low shape similarity. The assumption is that if two node shapes are different, it is very likely that there is a wrong PCRP. In this case, even the PCRP has a high node similarity with its counterpart in the model graph, we still reduce its contribution to the graph similarity. After introducing the weight factor, (23) becomes

$$Sim_G = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} E(i,j)\cdot S_{\mathrm{ANGSIM}}(\theta_{i,j}-\theta_{i0})\cdot Weight(i,j) \qquad (24)$$

Another element that helps the process is the relationship between graph nodes. In other methods, graph nodes are only points in 2-D or 3-D space and have only geometry relationships among them [5,28]. In our scenario, every node represents a region, i.e., a character of the object. These nodes have a certain pre-determined constraint.

For relationship checking, we have some basic assumptions.

1. *All regions in the model and object <u>have only above four relationships mentioned above</u>.*
2. *Transformations should not change the correspondent relationships, i.e. if for the regions R_1 and R_2, we have $RL(R_1, R_2) = RL(R_1', R_2')$. Otherwise, the matching fails. For example, contain and contained relationships are distinguishing characteristics, we do not expect that noise can drastically change such relations.*
3. *We require a basic relationship to retain the contiguous relation as an exception. We accept contiguous→separate and do further checking.*

We define four relationships between two regions – contiguous, contain, contained and separate. They are shown in Figure 18. Below is the relationship-checking table-2

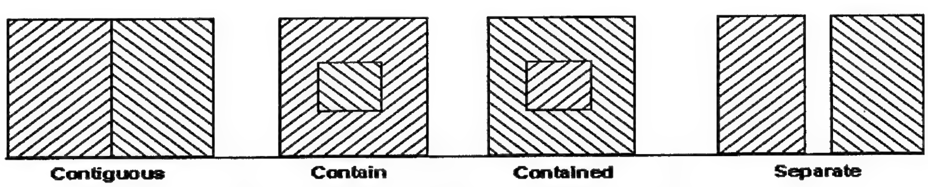


Figure 18: Four relations between two regions

Table 2: The relationship checking table T_{RE}.

Relationship – Model / *Relationship – Data image*	Contiguous	Contain	Contained	Separate
Contiguous	1	0	0	0
Contain	0	1	0	0
Contained	0	0	1	0
Separate	1/0	0	0	1

For any two regions R1 and R2, if their relationship in the image is r_{12} and in the model graph is r_{12}', then the similarity relationship between these two regions is

$$sim_{REL} = T_{RE}(r_{12}, r_{12}')$$

The relationship between two graphs is determined by relation checking at every region between any two connected graph nodes.

$$SIM_{REL} = \prod_{i=1}^{N}\prod_{j=1}^{N} T_{RE}(r_{i,j}, r'_{i,j}) \tag{25}$$

There is one relationship checking value undetermined in the table. If two regions are contiguous in the model graph but are separate in the generated graph, we do not classify it as violation of the relationship matching rule. Instead, we check the two regions spatial relationships further. Because of the noise and light conditions, small noise or a shadow could separate two regions in the generated graph while they are contiguous in the model. Figure 19 shows an example.

(a) (b)

Figure 19: (a), (b) are part of the model image and graph. The two regions are contiguous. (c), (d) are correspondent part form data image. Because of the shadow, those two regions are separate.

(c)

(d)

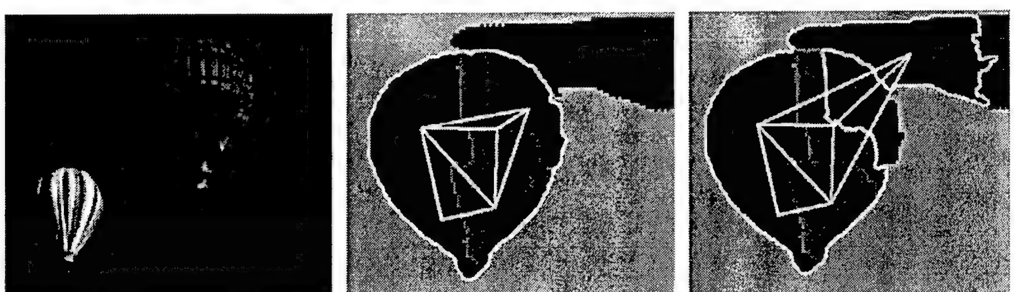
6. The *L-G* Graph Synthesis Method

Although the *L-G* graph method applies spatial constraints to graphs, it may still generate mismatches. For example, when two potential correspondent nodes from the same model node are too close in the examined image, both *L-G* graphs are similar to the model graph. In this case, the graph-comparing method cannot filter out all wrong graphs even after region relationship

checking. Figure 20 shows an example. Because the cloud color is similar to one of a balloon's part, the cloud becomes one PCRP. When we use this PCRP to replace the right node, a similar graph is generated. As a result, it could pass the graph similarity checking step, and due to its particular special location, it does not violate any region relationship. Thus, we need to find a way to exclude such cases and improve the matching accuracy. We still use the graph scheme, but now combined with the object shape information. We have used the shape similarity as a weight in previous matching steps. After the global graph matching, a potential region group has been found. This region group R_D has a similar spatial structure to the model's structure R_M. It is reasonable to expect that the object shape associated with R_D should be similar with the shape associated with R_M. If we can synthesize the region group to get the object shape, then the object shape similarity can be a criterion to determine the matching result.

In order to obtain the object shape from the region shape, a shape synthesis method based on the local graph is proposed. In this method, regions are merged together to generate the object shape. The *L-G* synthesis method is:

1) *Build the local graph from the current PCRP set.*
2) *Compare the local graphs.*
3) *If the graph is similar and correspondent spatial connectivity is consistent, start synthesizing processing.*
   3.1) Assign the first region to an active region.
   3.2) Select the next region. Find the common edge between it and the active region.
   3.3) If a common edge is found, synthesize the current region and the active region and assign the new region to active region.
   3.4) If all regions have been processed, synthesizing completes; otherwise, go to step (3.1)
4) *Compare the synthesized region with the model region. If similar, accept; otherwise, match fails.*



(a) (b) (c)

Figure 20: An example of a mismatch. (a) is the original data image. (b) and (c) show two graphs with one PCRP change. (b) and (c) have a similar graph and also the same region relationships.

6.1..Neighbor Region Searching by Local Graphs

If two regions are neighbors, they have at least one common border. Consider the issue shown in figure 19, we define the common edge with broad sense —if their borders have a common shape and the common parts are very close in Euclidean space, they are neighboring regions. If the correspondent common parts have distance greater than zeros, then these two regions are *pseudo-neighbors.*

Because we have introduced pseudo-neighbors, we cannot use the physical region border pixel location to compute the connectivity between regions. The region local graph represents its shape

in a high level. We use the local graph to find the common edge between regions. The borderline graphs of two regions are L_1, L_2.

$$L_1 = Ln_1 \cdot R^c_{1,2} \cdot Ln_2 \cdot R^c_{2,3} \cdot Ln_3 \cdot R^c_{3,4} \cdot \ldots Ln_{n-2} \cdot R^c_{n-2,n-1} \cdot Ln_{n-1} \cdot R^c_{n-1,n} \cdot Ln_n$$

$$L_2 = L'n_1 \cdot R^c_{1,2}{}' \cdot L'n_2 \cdot R^c_{2,3}{}' \cdot Ln_3 \cdot R^c_{3,4} \cdot \ldots Ln'_{m-2} \cdot R^c_{m-2,m-1}{}' \cdot Ln'_{m-1} \cdot R^c_{m-1,m}{}' \cdot Ln'_m$$

Suppose L_1 has n lines and L_2 has m lines. If L_1 and L_2 have a common edge, a part in L_1 must match with a part in L_2. We need to find if there is a partial matching and where it is in the two graphs. Now, we treat every element, line segment N and relationship a_c, in line graph (L_1, L_2) as the character of a string. Thus L_1, L_2 become two strings. Our next task is to find the matched sub-string in these two strings

$$L_1 = C(1) \cdot C(2) \cdot C(3) \ldots \cdot C(n)$$

$$L_2 = C'(1) \cdot C'(2) \cdot C'(3) \ldots \cdot C'(m)$$

Where $C(i)$, i=1,2,…,$max(m,n)$ is an element of a border-line graph. $C(i)$ could be a line N_i or a relation $R_{i,j}$. Suppose L_2 is longer than L_1, i.e. $m>n$ (if not, swap L_1, L_2). Slide L_1 through L_2 and compare "character by character". The comparison error function is defined as

$$S_{i,j} = \begin{cases} 1, & \text{if } f(i,j) = true \\ 0, & otherwise \end{cases} \tag{26}$$

$S_{i,j}$ is the result of comparing $C(i)$ from L_1 with $C'(j)$ from L_2; and $f(i,j)$ is the comparison function, defined as

$$f(i,j) = \begin{cases} false, & if\ C(i), C(j)\ have\ different\ type \\ true, & if\ C(i), C(j)\ are\ relations\ and\ (c, p, rm, rd) \approx (c', p', rm', rd') \\ true, & if\ C(i), C(j)\ are\ lines\ and\ ||C(i)| - |C(j)|| < \varepsilon_1, \\ & \quad |angle(C(i)) - angle(C(j))| < \varepsilon\ OR \\ & \quad C(i/j)\ is\ part\ of\ C(j/i) \\ false, & otherwise \end{cases} \tag{27}$$

before comparing, copy L_1 itself at the end of L_1. By doubling L_1, this method overcomes a starting-point breaking problem. The object border is closed. But a starting point will separate the border into head and end. If the starting point resides in the matching part, it breaks the substring. This is shown in figure 21.

Figure 22 shows an exmaple of finding regions common edges. The common edge between two selected regions are highlighted by "red" (double dot) lines.

These two regions are contiguous to each other, so they are real neighbors. Figure 23 shows the common edge between the regions shown in Figure 19. These two regions are neighbors in the model image, but separate in the examined image because of shadows. By using our common edge finding method, the common edges are detected and classified as pseudo-neighbors.

L_1 = abbdguiddfv.....odjfbdaa

L_2 = ...asdidnalbdaaabbdddf..

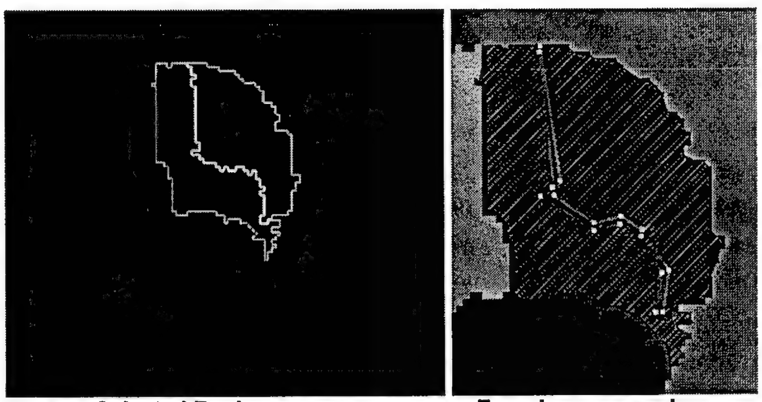
(a)

L_1 = abbdguiddfv....odjfbdaa abbdguiddfv....odj..

L_2 = ...asdidnalbdaaabbdf...

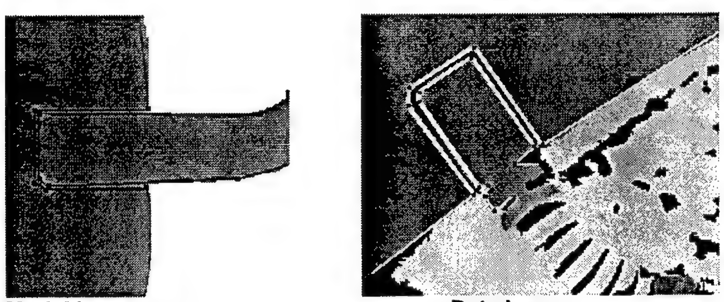
(b)

Figure 21: If the starting point of the local graph inside the match pattern, it will break the pattern. (a) shows such example. (b) shows how to fix this problem by cloning L_1.



Selected Regions **Found common edge**

Figure 22: Use local graph to find the common edge between regions



Model image **Data image**

Figure 23: The common edge is detected in model image and examined image respectively for example shown in Figure 19.

6.2..Region Synthesis

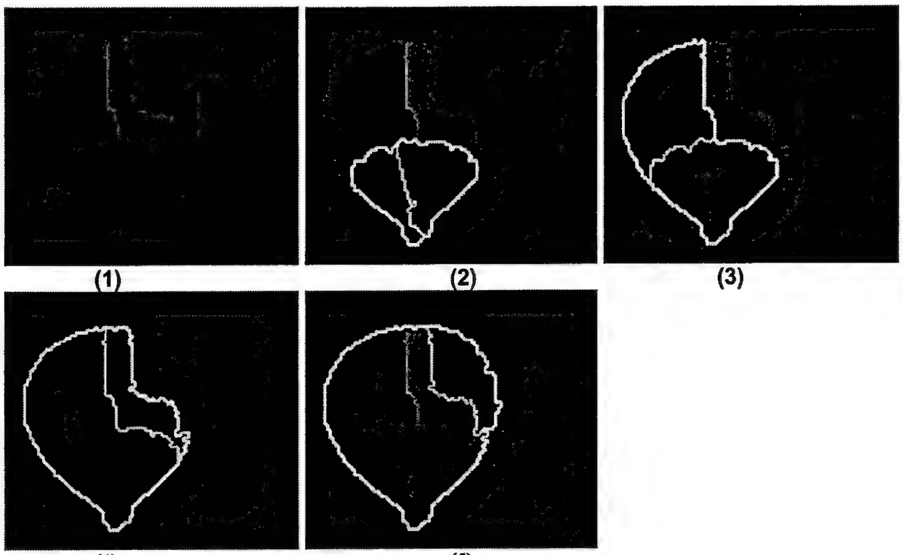
After a common edge is detected, the region synthesis process starts. The process merges all neighbor regions and extracts the whole object shape.

If there are two regions R_1 and R_2, the rules to synthesize them it to a new shape R_{12} are

- If $RL(R_1, R_2)$=contiguous, shape(R_{12}) = shape(R_1) Y shape(R_2)
- If $RL(R_1, R_2)$=contain, shape(R_{12}) = shape(R_1)

- If $RL(R_1, R_2)$=contained, shape(R_{12}) = shape(R_2)
- If $RL(R_1, R_2)$=separate, shape(R_{12}) = Φ

In order to improve the performance of the region synthesis process, we build the Table 3 to accelerate the finding of common edges process. In fact, we do not need to compute the relationship for every node pair. Sometimes, the relationship can be deduced from other relationships. In the merge process, the regions R_1 and R_2 are merged as a new region R_{12}. Then, the relationship between R_{12} and R_3 can be deduced somewhat from the relationships $RL(R_1,R_3)$ and $RL(R_2,R_3)$, where $RL(x,y)$ is the relationship between region x and region y. Table 3 summarizes all possible relations between R_{12} and R_3, given $RL(R_1,R_3)$ and $RL(R_2,R_3)$.
Figure 24 graphically shows the consecutive steps of the synthesis process.



(1) (2) (3)

(4) (5)

Figure 24: Illustration of the Region synthesis

Table 3. Deduce object12/object3 relationship after merging object1 and object2

R1→R2	R2→R3	R12→R3			
		Contiguous	Contain	Contained	Separate
Contiguous	Contiguous	•			
	Contain		•		
	Contained			•	
	Separate	•			•
Contain	Contiguous	•			
	Contain		•		
	Contained			•	
	Separate				•
Contained	Contiguous		•		
	Contain		•		
	Contained		•	•	

	Separate	•	•		•
Separate	Contiguous				
	Contain				
	Contained				
	Separate				

6.3 The Entire L-G Graph Matching Scheme

To formulate the matching measurement, an *L-G match error* $ERR_{L\text{-}G}$ is proposed as well. It is defined as

$$\boldsymbol{ERR_{L\text{-}G} = (1\text{-}ERR_{rel}) \cdot (ERR_G + ERR_{shape})} \qquad (28)$$

where the $\boldsymbol{ERR_{L\text{-}G}}$ is the total matching error between object and model, $\boldsymbol{ERR_{rel}}$ is the matching error of relationships among regions, $\boldsymbol{ERR_G}$ is the matching error between global graphs, and $\boldsymbol{ERR_{shape}}$ is the matching error between two objects' shapes.

Then, the entire *L-G* graph matching scheme is summarized as three cases,

1. **Object and model have only <u>ONE</u> region.**

Compare object's shape with model's shape.

One region corresponds to one node and there is no global graph for either model or object. Thus, $\boldsymbol{ERR_{rel}}$ is 1 and $\boldsymbol{ERR_G}$ is 0 because they are related to graph matching. In this case, the error $\boldsymbol{ERR_{L\text{-}G}}$ is simplified to only one term, i.e.,

$$\boldsymbol{ERR_{L\text{-}G} = ERR_{shape}}$$

If the shape matching error $\boldsymbol{ERR_{shape}}$ is less than a predetermined threshold Th_{shape},

i.e. $\boldsymbol{ERR_{shape}} < Th_{shape}$, then we consider that the object is similar to the model. Here the returned $\boldsymbol{ERR_{shape}}$ is the measure of similarity.

2. **Object and model have only <u>TWO</u> regions.**

Suppose the two object regions are R_{ob1} and R_{ob2}, and the two model regions are R_{md1} and R_{md2}.

The graph for two-region objects is a straight line. The global graph matching error $\boldsymbol{ERR_G}$ is eliminated. Now the Error function becomes

$$\boldsymbol{ERR_{L\text{-}G} = (1\text{-}ERR_{rel}) \cdot ERR_{shape}}$$

For two regions, there is only one relationship between them. Suppose the relationships are $Re_{ob} = RL(R_{ob1}, R_{ob2})$ and $Re_{md} = RL(R_{md1}, R_{md2})$. If the relationship Re_{ob} and Re_{md} do not match, i.e., $Re_{ob} \neq Re_{md}$, the matching process fails; otherwise, continue.

Merge R_{ob1}, R_{ob2}. $\quad R_{ob1}$ Y $R_{ob1} \rightarrow R_{ob12}$.

Merge R_{md1}, $R_{md2}R_{md1}$ Y $R_{md1} \rightarrow R_{md12}$.

If the shape matching error is less than the preset threshold Th_{shape}, i.e. $\boldsymbol{ERR_{shape}} < Th_{shape}$, then the object is similar to the model.Thus, the returned $\boldsymbol{ERR_{shape}}$ is the measure of similarity.

3. **Object and model have more than two regions.**

Suppose object regions are

$S_{ob} : \{R_{ob1}, R_{ob2}, \dots , R_{obm}\}$, m is the regions count.

Model regions are

$S_{md} : \{R_{md1}, R_{md2}, \dots , R_{mdm}\}$, m is the regions count.

Now the Error $\boldsymbol{ERR_{L\text{-}G}}$ has all three terms,

$$\boldsymbol{ERR_{L\text{-}G} = (1\text{-}ERR_{rel}) \cdot (ERR_G + ERR_{shape})}$$

Let a relationship set be

$Re_{ob} = \sum RL(\{S_{ob}(i), S_{ob}(j)\}, i,j=1 \dots m)$

$Re_{md} = \sum RL(\{S_{md}(i)\ S_{md}(j)\}, i,j=1 \dots m)$

Then, the relationship set Re_{ob} and Re_{md} are compared based on the Table 3. If all region (node) relationships comply with the condition in Table 3, then the relationship error $\boldsymbol{ERR_{rel}}$ returns 0; otherwise it returns 1 and the process ends.
Then, build the global graph from the object and model region set S_{ob} and S_{md},

$$GH_{ob} = DE(S_{ob})$$

and

$$GH_{md} = DE(S_{md})$$

where $DE(\cdot)$ is the *Global Graph* operator. The global graph error $\boldsymbol{ERR_G}$ is obtained by comparing GH_{ob} and GH_{md}. The global graph definition requires that not all nodes lie on a straight line. If it happens, we cannot create a global graph from that node set. This is a pre-condition of $\boldsymbol{ERR_G}$. We check this condition prior to computing the graph error. If not all nodes are one straight line, the $\boldsymbol{ERR_G}$ is computed based on equations (22) and (23). We notice that the object region could be affected by many factors, such as view angle, data acquiring method and equipment, light condition, *etc.* A Human's perception system generally has a large tolerance to the region deformation, but is keen to shape deformation. Thus, a relatively large threshold Th_{graph} is set for $\boldsymbol{ERR_G}$. If the $\boldsymbol{ERR_G}$ is bigger than Th_{graph}, it means that the current object has a very different structure from the model. We will not consider them as similar and quit the current step. Otherwise, the error term $\boldsymbol{ERR_G}$ is saved.Then, we proceed to the next matching step.
In the third step, current object regions are merged based on their internal relationships.

Merge object set, $SHP_{ob} = \Upsilon\ \{S_{ob}\}$

Merge model set, $SHP_{md} = \Upsilon\ \{S_{md}\}$

The region synthesis method is described in the section above. In fact, the model is saved in the database prior to the matching process. It's shape can be synthesized by adding it into the database. Thus, we need only to do it once and compare it with any object shapes. This will improve the matching performance.
After synthesizing, we compute the shape similarity between SHP_{ob} and SHP_{md}. Basically, the single region comparing method (described in a previous section) is applied to them. If the shape matching error is less than the predetermined threshold Th_{shape}, i.e., $\boldsymbol{ERR_{shape}} < Th_{shape}$, then the object is similar to the model. The returned $\boldsymbol{ERR_{shape}}$ is saved.
If all three terms are computed and fall into the acceptable range, the object is located and recognized.
The total similarity measurement is computed by (28). The smaller the $\boldsymbol{ERR_{L\text{-}G}}$ is, the more similar the object and model are. If, however, the matching process fails, the generated graphs may be saved in a separated part of the database for possible future use. This part of the matching process may represent the "learning" part of the recognition by providing information about deformed objects where the corresponding graphs do not match with the models in the regular graph-database [33].

7. Conclusions

This paper has presented an object recognition method mainly based on Wavelets, L-G graphs and the synthesis of the regions that compose an object by using their graph representations. The methodology belongs to the model based recognition, where models of objects exist in a graph based database. The results are accurate as long as the extracted candidate object L-G graph has a satisfactory matching with an object-model in the LG graph database. An important point of the object recognition process is the synthesis of the object regions by using the LG graph and neighborhood criteria, such as adjacent lines. This feature helps for the extraction and generation of the most accurate object model that has to be matched against the existing database. Another important feature this methodology is its learning capabilities. In particular, the learning scheme here is based on the extraction and generation of object L-G graph models that have no matching with the ones in the database. For instance, if finally an object LG graph model is produced by

this method (by iteratively selecting different neighbor regions in various combinations) with no satisfactory matching acceptance, this particular LG graph model is saved into the database and classified as a new object (Oj). This new object (Oj) stays in the database as along as no better version is extracted from the same scene for it. This may happen if a camera is capturing images from a certain parking lot with different weather conditions. This means that the pixels of the same image extracted from the same scene are now functions and no just single values. In other words, illumination may create artifacts and a "new LG graph object-model (Oj)" to be saved in the database. Later, when the illumination effect change, a new version of the same object may be recognized and an appropriate correction or replacement, of the object (Oj) will take place. On the other hand, if no new version of the object (Oj) is found, then this object is new and a name is given to it. In other words, the system learns new objects and associated them in an object-category with similar features in the database. Details for the learning process are available in [33].

References

[01] D.P. Huttenlocher, G. Klanderman and W.J. Rucklidge, "Comparing Images Using The Hausdorff Distance", *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 25, no. 9, pp. 850-863, 1993.

[02] A. Sanfeliu and King-Sun Fu, "A Distance Measure Between Attributed Relational Graphs For Pattern Recognition", *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 13, no. 3, pp. 353-362, 1983.

[03] Y. Amit and A. Kong, "Graphical Templates For Model Registration", *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 18, no. 3, pp. 225-236, 1996.

[04] Ronald Alferez and Yuan-Fang Wang, "Geometric And Illuminating Invariants For Object Recognition", *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 21, no. 6, pp. 505-536, 1999.

[05] A. D.J. Cross and E. R. Hancock, "Graph Matching With A Dual-Step EM Algorithm", *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 20, no. 11, pp. 1236-1253, 1998.

[06] N.. Bourbakis, "Generating 2-D Space Maps From Unknown Environments", *International Journal Of Pattern Analysis and Artificial Intelligence*, vol. 13, no. 3, pp. 297-318, 1999.

[07] S. E. Sclaroff, "Model Matching: A Method For Describing, Comparing And Manipulating Digital Signals", *PhD Thesis*, MIT, 1995.

[08] E.. El-Kwae and M. R. Kabuka, "A Robust Framework For Content-Based Retrieval By Spatial Similarity In Image Database", *ACM Transaction On Information Systems*, vol. 17, no. 2, pp. 174-198, 1999.

[09] A. Celentano and E. Di Sciascio, "Feature Integration And Relevance Feedback Analysis In Image Similarity Evaluation", *Journal of Electronic Imaging*, vol. 7, no. 2, pp. 308-317, 1999.

[10] A. Pentland and S. Sclaroff, "Closed-Form Solutions For Physically Based Shape Modeling And Recognition", *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 13, no. 7, pp. 715-729, 1991.

[11] Q. M. Tieng and W. W. Boles, "Recognition Of 2-D Object Contours Using The Wavelet Transform Zero-Crossing Representation", *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 19, no. 8, pp. 910-916, 1997.

[12] M. Werman and D. Weinshall, "Similarity And Affine Invariant Distances Between 2-D Point Sets", *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 17, no. 8, pp. 810-814, 1995.

[13] E. Kubicka, G. Kubicki and I. Vakalis, "Using Graph Distance In Object Recognition", *1990 ACM Eighteenth Annual Computer Science Conference Proceedings, ACM, New York, NY*, pp. 43-48, 1990.

[14] L. Jia and L. Kitchen, "Object-Based Image Similarity Computation Using Inductive Learning Of Contour-Segment Relations", *IEEE Trans. On Image Processing*, vol. 9, no. 1, pp. 80-88, 2000.

[15] J. T.L., Wang, B. A., Shapiro, D. Shasha, K. Zhang and K. M. Currey, "An Algorithm For Finding The Largest Approximately Common Substructures Of Two Trees", *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 20, no. 8, pp. 889-895, 1998.

[16] X. Yi and O. I. Camps, "Line-Based Recognition Using A Multidimensional Hausdorff Distance", *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 21, no. 9, pp. 901-919, 1999.

[17] S. Abbaui, F. Mokhtarian and J. Kittler, "Curvature Scale Space Image In Shape Similarity Retrieval", *Multimedia-Systems*, vol. 7, no. 6, pp. 467-476, 1999.

[18] H. D. Tagare, F. M. Vos, C. C. Jaffe, J. S. Duncan "A Spatial Relation Between Parts For Similarity Of Tomographic Section", *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 17, no. 9, pp. 880-893, 1995.

[19] W Wan and J A. Ventura, "Segmentation Of Planar Curves Into Straight-Line Segments And Elliptical Arcs", *Graphical Models and Image Processing*, vol. 59, no. 6, pp. 484-494, 1997.

[20] G. A. W. West and P. L. Rosin, "Techniques For Segmenting Image Curves Into Meaningful Descriptions", *Pattern Recognition,* vol. 24, no. 7, pp. 643-652, 1991.

[21] N. Ahuja, B. An and B. Schachter, "Image Representation Using Voronoi Tessellation", *Computer Vision, Graphics and Image Processing*, vol. 29, pp. 286-295, 1985.

[22] N. Ahuja, "Dot Pattern Processing Using Voronoi Neighborhoods", *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 4, no. 3, pp. 336-342, 1982.

[23] K. Arbter, W. E. Snyder, H. Burkhardt and G. Hirzinger, "Application of Affine-Invariant Fourier Descriptors to Recognition of 3-D Object", *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 12, no. 7, pp. 640-646, 1990.

[24] N. Bourbakis, Emulating human visual perception for measuring differences in images using an SPN graph approach, *IEEE Trans. on Systems, Man and Cybernetics*, 32,2,191-201, 2002

[25] Hu MK, "Visual Pattern Recognition by Moments Invariants", *IRE Transaction of Information Theory*, IT. 8, pp. 179-187, 1962.

[26] T. Pavlidis, "Structural Pattern Recognition", *Berlin, New York: Springer-Verlag,* 1977.

[27] S. Ranade and A. Rosenfeld, "Point Pattern Matching By Relation", *Computer Society Tech. Rep.*, TR-702, 1978.

[28] M. Celenk, "Relational Graph Representation Of Color Images For Model-Based Matching Using Relational Distance Measurement", *Visual Information Processing IV Proc. SPIE,* pp. 229-240, 1995.

[29] S. Sclaroff and A. P. Pentland, "Generalized Implicit Functions For Computer Graphics", *ACM Computer Graphics*, vol. 25, no. 4, pp. 247-251, 1991.

[30] X. Yuan, "Recognition of 3D objects", PhD Thesis SUNY, 2002 *KDE*

[31] N. Bourbakis, "A Rule-Based Scheme For Synthesis of Texture Images", *Int. IEEE Conf. on systems, Man and Cybernetics*, Fairfax, VA, pp. 999-1003, 1987.

[32] N. Bourbakis, "Extraction, Tracking and Recognition Of Targets In Sequences Of Images", *Int. Journal AIT*, vol. 11, no. 4, 2002.

[33] N. Bourbakis, "Recognition of deformed objects using 3D L-G graphs", AIIS report, 2003.

[34] H-T Sheu and W-C. Hu, "Multi-primitive Segmentation Of Planar Curves-A Two-Level Breakpoint Classification And Tuning Approach", *IEEE Trans. on Pattern Recognition and Machine Intelligence,* vol. 21, no. 8, pp. 791-797, 1999.

[35] D. Fotakis and N. Bourbakis, "A Heuristic Scheme For The Recognition Of Progressive Digital Straight Lines With Unevenness", *Int. IEEE Workshop on LFA, MD*, pp. 176-184, 1988.

[36] N. Bourbakis and D.Goldman, Recognition and acquisition of digital line segments with unevenness with applications to recognition of handwritten characters and fingerprints, IFAC IJEAAI,12,273-279,1999

[37] D.Hearn and M. Baker, "Computer Graphics, C version", *Upper Saddle River, NJ: Prentice Hall*1997.

[38] Q. Qing and Z. Yang, "Practical Wavelet Analysis", *Xian Electronics Tech., Xian, China,* 1995.

[39] Zahn CT, "Using The Minimum Spanning Tree To Recognize Dotted And Dashed Curves", *International Computing Symposium*, North-Holland, Amsterdam, Netherlands, pp. 381-387, 1974.

[40] H. Bunke, "Error Correcting Graph Matching: On The Influence Of The Underlying Cost Function", *IEEE Trans. on Pattern Recognition and Machine Intelligence,* vol. 21, no. 9, pp. 917-922, 1999.

[41] L. Davis, "Shape matching suing relaxation techniques", *IEEE Trans. on Pattern Recognition and Machine Intelligence,* vol. 1, pp. 60-72, 1979.

Acknowledgement

The AIIS personnel and the WSU research group express their great appreciation to Mr. S Borek and Dr. Herklotz for their generous support and professional guidance for achieving our goals proposed in the AFRL-STTR Phase I proposal. Also we like to thank any other AFRL personnel Involved in the process.